

AD845521

AD 845521

DDG FILE COPY



COAST GUARD

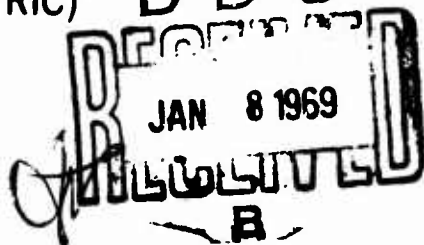
SHIP DESIGN AND MAINTENANCE COMPUTER PROGRAM

ICEBREAKER PROPULSION SYSTEM SIMULATION

(DIESEL - ELECTRIC) D D C

MAY 1968

JAN 8 1969



U.S. COAST GUARD HEADQUARTERS

OFFICE OF ENGINEERING

X NAVAL ENGINEERING DIVISION

PROGRAM NO. ENE-11

UNCLASSIFIED		DATE SECTION	<input checked="" type="checkbox"/>
JUSTIFICATION		DATE SECTION	<input checked="" type="checkbox"/>
BY			
OBTAINING/AVAILABILITY CODES			
DIST.	MAIL	and/or	SPECIAL
MZ			

TCD New source y AD 845 520;
 CG, WDC. Off of Eng (404 342).
 Lower echelons x-vap'd to
 Keep repts together

Allen Kuhn
 9 Jan 68

DEPARTMENT OF TRANSPORTATION

U.S. Coast Guard Headquarters
Washington, D. C. 90591

SHIP DESIGN AND MAINTENANCE COMPUTER PROGRAM

Title: ICEBREAKER PROPULSION SYSTEM SIMULATION (DIESEL-ELECTRIC)

APPROVAL

The contents of this documentation are in accordance with good engineering design and computer programming practices.

<u>TITLE</u>	<u>BRANCH/PROJECT</u>	<u>SIGNATURE</u>	<u>DATE</u>
Project Officer/Engineer	ENE-9A	J. W. Lamin	27 May '68
Computer Aided Design Coordinator	ENE-9A	J. W. Lamin	27 May '68
Chief	ENE-9A	J. B. Schumacher	27 May '68

This documentation is correct and complete and the program is coordinated with related work in other branches.

	<u>SIGNATURE</u>	<u>DATE</u>
Coordinator Computer Application - Naval Engineering	J. B. Schumacher	27 May '68

LIMITATION OF LIABILITY

The U. S. Government will incur no liability resulting from use of this computer program by organizations or persons other than the U. S. Coast Guard.

This document is subject to special export controls and transmittal to foreign governments or foreign nationals may be made only with prior approval of the Commandant (Eg), U. S. Coast Guard.

(See form 173)

Computer
Documentation
J. W. Lamin

TABLE OF CONTENTS

<u>Section</u>	<u>Description</u>	
PART I	IDENTIFICATION	1
1.	Title	1
2.	Brief Description	1
3a.	Author	1
b.	Date	1
c.	Documentation	1
4.	Language.	1
5.	Machine	1
6.	Security Classification	1
7.	Estimated Running Time	1
PART II	PURPOSE AND METHOD	2
1.	Purpose	2
2.	Method and Theory	2
	a. Method	2
	b. Theory	2
	c. References	3
PART III	RESTRICTIONS	4
1.	General Restrictions	4
2.	Tapes Required	4
3.	Nonstandard Hardware	4
4.	Maximum Array Sizes.	4

<u>Section</u>	<u>Description</u>	<u>Page</u>
PART IV	NONSTANDARD OPERATING INSTRUCTIONS	5
1.	Special Operating Instructions	5
	a. General	5
	b. Documentation Procedures	5
	c. Problem Solving.	6
	d. Program Alterations	8
	e. Plotter Output	8
2.	Error Corrections	9
3.	Restart Instructions	9
4.	Program Changes.	9
PART V	DATA PREPARATION	13
1.	Input Data	13
2.	Output Data from Sample Run	89
APPENDIX		101
1.	Symbol Table	102
2.	System Flow Charts	112
3.	Mainline Programs and Subroutine Flow Charts.	116
4.	Mainline Programs and Subroutine Documentation	151
5.	Program Listings	201

PART I - IDENTIFICATION

1. Title: Icebreaker Propulsion System Simulation (Diesel-Electric).
2. Brief Description: This program solves numerically the set of simultaneous non-linear differential equations which describe the dynamic behavior of the complete propulsion system (including vessel dynamics) of a diesel-electric icebreaker during maneuvering. The program also contains a means of subjecting the propeller to ice impact loadings of the user's making. At the option of the user, input data can readily be printed for documentation purposes and output data can be plotted.
- 3a. Author: Jack W. Lewis, Lieutenant, U.S. Coast Guard
- b. Date: May 1968
- c. Documentation: Prepared by Computer Usage Development Corporation, Washington Scientific Office.
4. Language: FORTRAN IV (IBM 1130 version)
5. Machine: IBM 1130 with disk (plotter desirable but not necessary).
6. Security Classification: UNCLASSIFIED
7. Estimated Running Time: Will vary directly with length of maneuver and time step employed. Using time steps of .08 seconds will require approximately 7 minutes computation time for 15 seconds of ship time. After initial propulsion plant transients have disappeared, generally within 10 to 20 seconds after the start of a maneuver, the time step can be increased to .15 to .20 seconds thus reducing computation time.

PART II - PURPOSE AND METHOD

1. Purpose: The purpose of this program is to simulate transient conditions within a diesel electric propulsion system for an icebreaker. The program will simulate various conditions to which an icebreaker may be subject, including crash reversals and accelerations under free route conditions, ice ramming, and subjection of propellers to ice encounters. Use of the program will permit designers to realistically study the effect of the transient conditions on design parameters and to optimize design selection of equipment for new icebreakers as required by the U. S. Coast Guard.
2. Method and Theory:
 - a. Method: The construction of mathematical models for complete ship propulsion systems involves writing and solving a large number of coupled non-linear differential equations. This computer model for solving these equations describes as accurately as possible the complete dynamics of all components of the icebreaker's propulsion system. Numerical integration techniques are used to simultaneously solve this complete set of equations and to present the resulting data as a time history of key variables both on a printer and graphic plotter.
 - b. Theory: The theoretical approach used in developing the model was to derive the equations and program them for computer solution. The computer solution for the USCGC GLACIER was compared to sea trial data for this ship. Complete theory and results of initial test runs is presented in references 1, 2 and 3 below.

c. References:

ICEBREAKER PROPULSION SYSTEM SIMULATION, U. S. Coast Guard Publication, 1967, by Lt. J. Lewis, U. S. Coast Guard, E. Scoville, and LCDR E. J. Lecourt, U. S. Coast Guard.

- 1) Phase I - Simulation of USCGC GLACIER.
- 2) Phase II - Dynamics of AC-DC Icebreaker Propulsion System.
- 3) Phase III - Propulsion System Behavior During Propeller - Ice Interactions.

PART III - RESTRICTIONS

1. General Restrictions: See other restrictions.
2. Tapes Required: None
3. Nonstandard Hardware: A Calcomp plotter is desirable but not necessary to run a simulation. If a plotter is not available, the GRAPH mainline routine and its four subroutines -GRID, DATA, VAPLO, and PRINT - must not be compiled nor any attempts be made to execute the GRAPH routine (data switch 2 must therefore remain OFF).
4. Maximum Array Sizes: The only size restriction of any concern is the limit of 600 records that can be stored on the disk during a run. A disk write interval time step must therefore be such as to not cause more than 600 records to be written on the disk.

PART IV - NONSTANDARD OPERATING INSTRUCTIONS

1. Special Operating Instructions:

- a. General: This program has been written in a manner which places the engineer in the computation loop. This type of operation was chosen in lieu of batch processing because of 1) the experimental nature of the problem frequently requires decision making during runs and 2) the IBM 1130 computer is quite adaptable to this type of operation. Considerable use is made of the keyboard console for the decision making process.
- b. Documentation Procedures: The sole purpose of the mainline program GENIS is documentation of each run. Although it is somewhat expensive in machine time to swap mainline programs in and out of core storage, the advantages of being able to document each run overrides.

Documentation is handled both in a decision making process at the keyboard console and by GENIS itself. In other words, certain input data and headers will automatically be printed as output whereas other input may or may not be printed depending on the console selector switches turned on. On the very first run switch 9 should be turned on in order that the serial number disk file can be initialized. This switch should then be left in the off position until it is desired to reinitialize the serial number file. Printed instructions will appear on the keyboard console to guide the user through each run.

- c. Problem Solving: The primary problem involved in the solution of differential equations by numerical methods is the selection of the time step. If the time step is too large the numerical integration method may lose the solution and "blow up." If the step is too small solution time can be prohibitive. As a general rule it is best to make several runs at first with different time steps to compare accuracy and solution time. During the first parts of any maneuver (usually the first 10 to 20 seconds) experience with this program indicates time steps around .06 to .08 seconds to be about right.

It is possible to interrupt the solution of a problem at any time by turning on console selector switch zero. In addition to stopping the process of integration, turning this switch on calls in subroutine FINIS which allows the user to select a number of options concerning what he wants to do next. For example if it is desired to change the time step, switches 3 and 5 would be turned on (make sure switch zero is off) and the push button marked "START" depressed. A message "() TIME STEP" will be typed at which time the user types in the new value of the time step. The key marked "EOF" (end of field) is depressed and the program continues the problem solution using the updated value of the time step.

Experience has shown that the time constant associated with the armature circuit governs the size of the time step. Generally speaking, after the armature circuit current stops fluctuating greatly (usually after 10 to 20 seconds of a maneuver) the time constant of this circuit can be increased without introduction of errors. Increasing the time constant allows larger time steps and hence decreases solution

time. Increasing this time constant can be readily accomplished by increasing the value of the armature circuit inductance (XIA). A value of .008 henries allowed a solution to be accomplished with time steps of .20 seconds in one case with no noticeable difference in the solution obtained by not changing the inductance (.0002 henries) and keeping the time step at .08 seconds.

It is a simple programming problem to change subroutine FINIS to include additional options desired by the user. Switches 8 through 15 are still available for this purpose.

Subroutine CNTRL was written with the idea that the user could add a library of maneuvers and data changes through the use of the variable ICNTR. For example a "Flank Power Ahead to Flank Power Astern" maneuver could be called using ICNTR equal to 2 and setting predetermined initial conditions into the program in addition to the ordered bridge controller position.

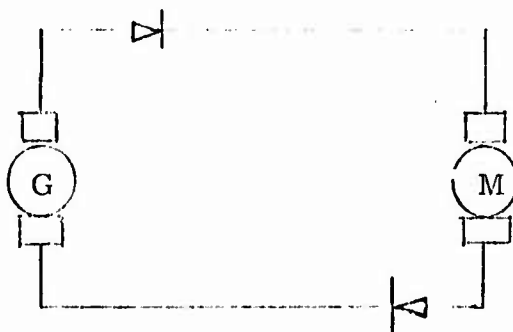
It is often easier to obtain the necessary initial conditions for a particular maneuver by using the simulation. For example, suppose it is desired to conduct a "Flank Power Ahead to Flank Power Astern" maneuver. The initial conditions at "Flank Power Ahead" are needed. These could be calculated roughly by hand, fed into the simulation and then by keeping the simulation at "Flank Power Ahead" allow the simulator to find the correct initial conditions. Once found, a new set of initial condition cards can be created by using switch 1 option of FINIS.

- d. Program Alterations: Altering the program has already been briefly mentioned above and will be further expanded upon here. An attempt was made to place components that may possibly be changed by a user into separate subroutines. This should facilitate changing components from one design to another. For example to change diesel engines, subroutine QDEV would have to be changed. If idle speeds or max speeds are changed it would also be necessary to make changes in subroutine CON2 which is the bridge controller command generator subroutine.

It is hoped that the program contains sufficient comments so that a user with the aid of this documentation can make major or minor changes to the program to solve his specific problem

- e. Plotter Output: Mainline program GRAPH is called into operation by turning on selector switch 2 after a program interrupt. Plotting and labeling the solution data is the purpose of this program. Before the data is plotted the user must supply the program with three essential pieces of information, i. e., 1) which variables the user wishes to be plotted 2) whether or not a grid and labeling is to be constructed and 3) any titling information. Data on variables to be plotted can be entered in any order; however, leave no spaces and insure the data is right justified. Drawing the grid requires time; therefore, during parametric investigation it is often best to first draw one grid and then use this for the remaining problems. Titling information may be omitted by depressing the key "EOF" three times. After the data is plotted mainline program GENIS is reloaded automatically.

2. Error Corrections: None
3. Restart Instructions: None
4. Program Changes: Each of the subroutines are written in modular form and in FORTRAN. This is done to allow the user options of changing the routines to simulate special situations. As an example of the type of changes that might be made let us suppose the armature circuit looked as follows:



This circuit would only allow unidirectional current flow. Following is a modified subroutine CALVR which would simulate this situation - i. e. if $EC > EG$ then curvature circuit = 0.0.

CALV0520
CALV0530
CALV0540
CALV0550
CALV0560
CALV0570
CALV0580
CALV0590
CALV0600

CALCULATE GENERATOR FLUX, INDUCTANCE, AND
TIME CONSTANT

CALL FLUXG
EG=60.*CFG*XNPV

CALCULATE MOTOR FIELD FLUX, INDUCTANCE, AND
TIME CONSTANT

CALL FLUXV
EC=60.*CFM*XNP

IF(EG-EC) 400,250,250

400 AI=0.0

FI=EC

CG=0.0

CV=0.0

GO TO 251

250 CG=7.04*CFG*AI

CV=7.04*CFM*AI

ED=EG-(RG/XGEN)*AI

CALCULATE MOTOR FIELD CURRENT

CHECK FOR REVERSE DIRECTION

251 IF(THETA) 5,6,6

5, GO TO(7,8),:PLUG

SAVE INITIAL TIME AND MOTOR FIELD CURRENT

TTNET

FV1IC=FV1

IFLUG=2

8 ABLE=1./EXP((T-TTN)/(TAUA))

BAKER=1./EXP((T-TTN)/(TAUB))

FV1=FM1IC+SV1RV/(TAUA-TAUB)*(TAUA*(2.-ABLE)-TAUB*(2.-BAKER))

GO TO 34

IF(EB-C1) 10,10,20

20 AIHAT=XK1*(ER-C1)

FV1=FM1AX-FIP

GO TO 30

AIHAT=0.

30 IF(FV1-84.0) 31,32,32

31 FM1=84.0

32 IF(FV1-F1MAX) 34,34,33

33 FV1=F1MAX

CFRM-CSHFR*ABS(XNP)*XNP

CALCULATE SHIP'S RESISTANCE, WAKE AND THRUST
DEDUCTIONS

CALCULATE ADVANCE VELOCITY OF PROPELLER

CALCULATE IDEAL PROPELLER TORQUE AND
THRUST PER PROPELLER

CALCULATE TOTAL PROPELLER THRUST

OP=QPI

TP=TDSD*XPROI*XXX

CHECK IF SHIP HAS ENCOUNTERED ICE AND SET
ICE TORQUE TO ZERO IF NOT

IF(CT-TICE)50,51,51

CALV0630
CALV0640
CALV0650

CALV0670
CALV0680
CALV0690

CALV0700
CALV0710
CALV0720

CALV0730
CALV0740
CALV0750

CALV0760
CALV0770
CALV0780

CALV0790
CALV0800
CALV0810

CALV0820
CALV0830
CALV0840

CALV0850
CALV0860
CALV0870

CALV0880
CALV0890
CALV0900

CALV0910
CALV0920
CALV0930

CALV0940
CALV0950
CALV0960

CALV0970
CALV0980
CALV0990


```

50 Q1=0.0
   GO TO 52
C
51 CALL Q1CE
C
52 CALL STATE
   RETURN
   END

```

```

CALCULATE ICE TORQUE OF PROPELLER
CALCULATE DERIVATIVES

```

```

CALV1000
CALV1010
CALV1020
CALV1030
CALV1040
CALV1050
CALV1060
CALV1070

```


PART V- DATA PREPARATION

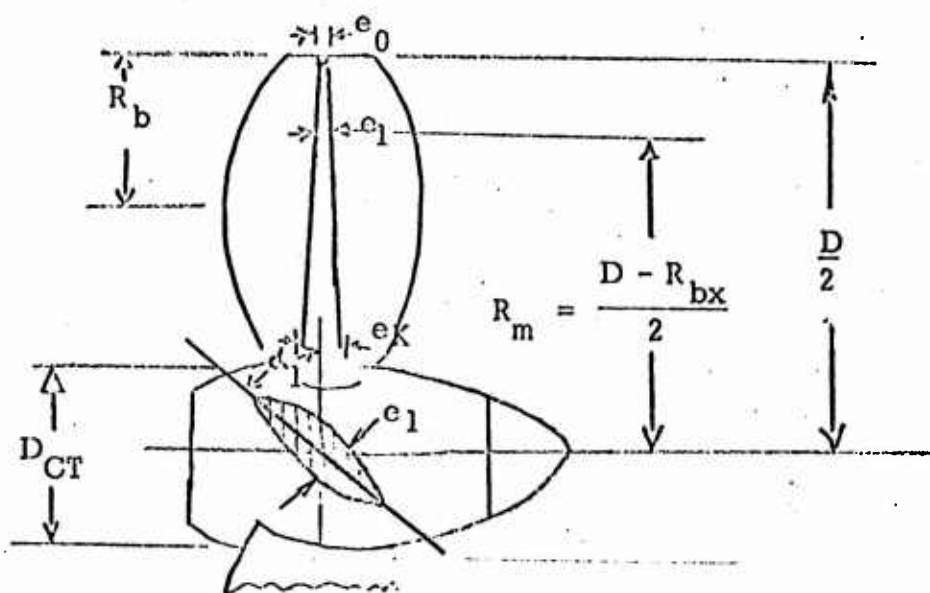
1. Input Data: The following indicates in detail how each data card is prepared and from where the data is gathered or calculated.

Each page or section of the report discusses individual data cards, preceded by an example data card prepared for the simulation discussed in references 1, 2 and 3.

PROPELLER DATA

IEK = Thickness of propeller blade at root (Feet)

DCT = Diameter of hub (Feet).



0.0	.000	.000	-0.1	.000	.000	INITIAL
J(I)	B(I)	BB(I)	J(I+1)	B(I+1)	BB(I+1)	
00000000	0	00000000	00000000	00000000	00000000	0
11111111	1	11111111	11111111	11111111	11111111	1
22222222	2	22222222	22222222	22222222	22222222	2
33333333	3	33333333	33333333	33333333	33333333	3
44444444	4	44444444	44444444	44444444	44444444	4
55555555	5	55555555	55555555	55555555	55555555	5
66666666	6	66666666	66666666	66666666	66666666	6
77777777	7	77777777	77777777	77777777	77777777	7
88888888	8	88888888	88888888	88888888	88888888	8
99999999	9	99999999	99999999	99999999	99999999	9

ADC 1131

Data Cards 2 - 33

FORMAT 2 (10x, 20x10.4)

ELEMENTS

J = Non dimensional speed

$$J = \frac{V_a}{n_p D}$$

where 'V_a' = speed of advance in feet/second

n_p = propeller speed in revolutions per second

D = as previously defined.

B = Thrust coefficient (non-dimensional) = K_T

$$K_T = \frac{T}{\rho(n_p)^2 D^4}$$

where ρ = mass density of water $\left(\frac{\text{lbs-sec}^2}{\text{ft}^4}\right)$

T = Thrust in lbs.

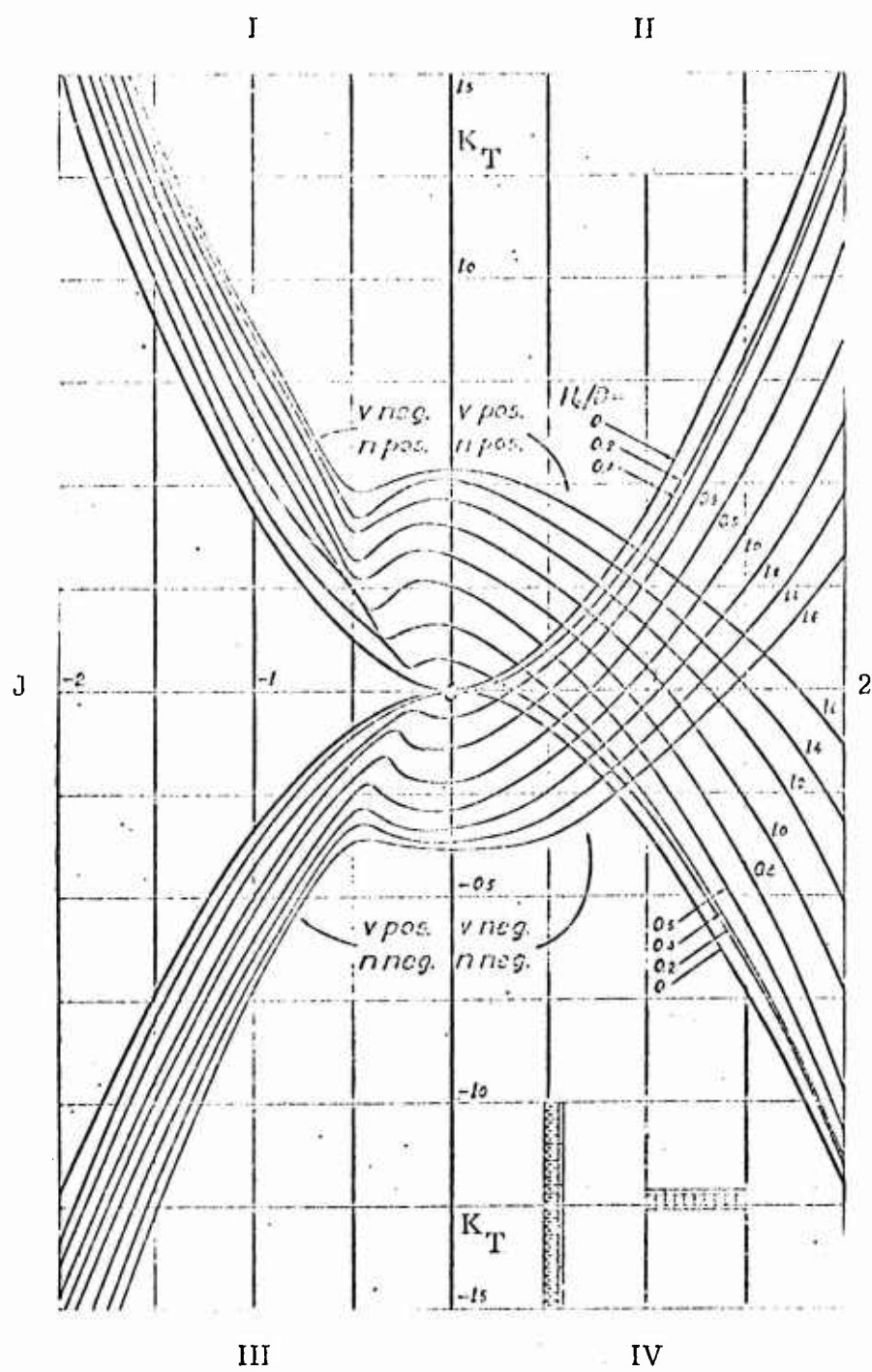
BB = Torque coefficient (non dimensional) = K_Q

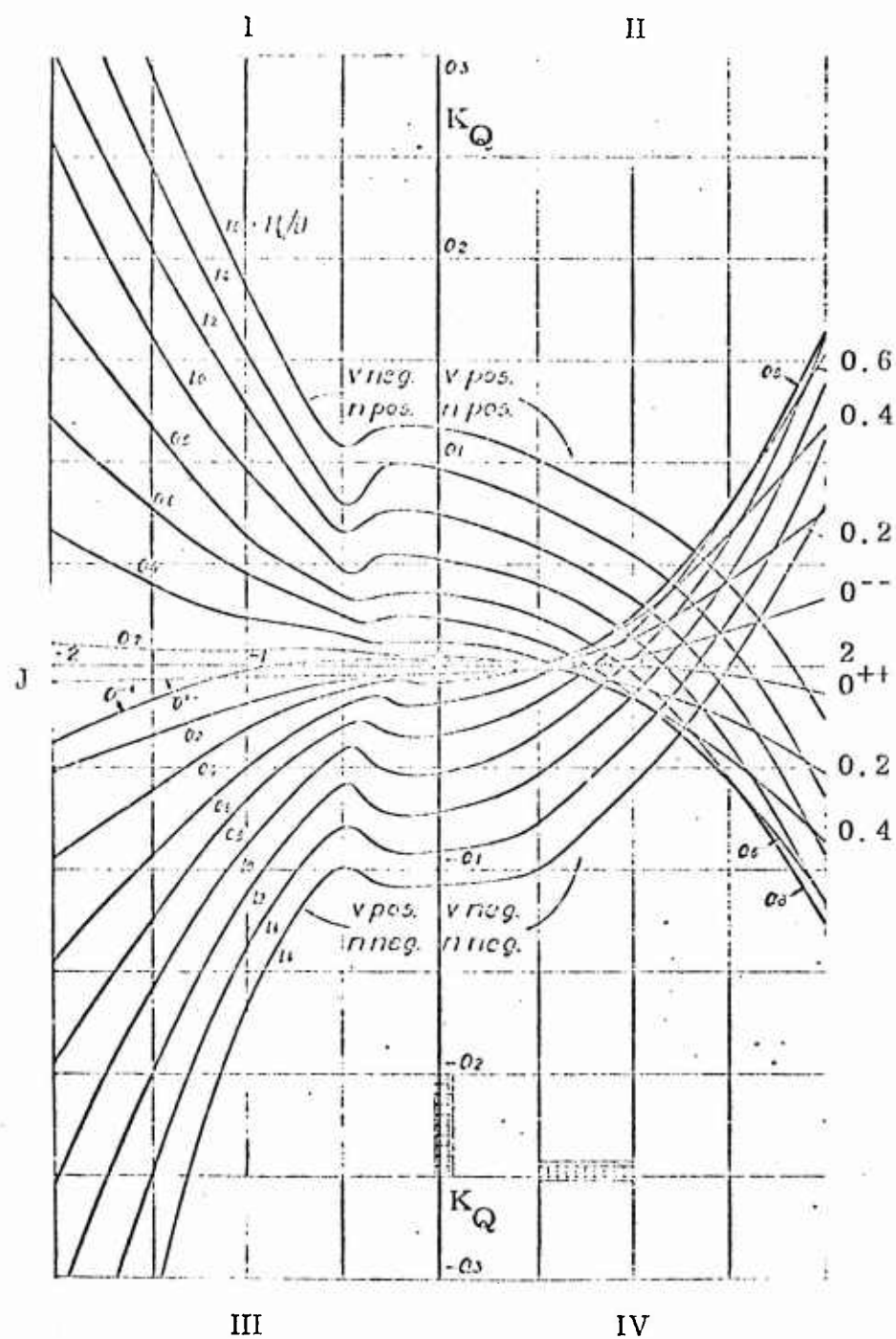
$$K_Q = \frac{Q_p}{\rho(n_p)^2 D^5}$$

Q_p = Torque in foot - pounds

These elements are read from standard propeller curves (Miniovich or Nordstrom) in the order of quadrants I, II, III, IV as shown on the following figures. Note that the J spacing must be as shown on the example cards as the J value is not read, it is assumed by the program.

Note: Written description continues on page 33.





DATA0012

100

DATA0013

144 5099

DATA0014

1-4 3 2-1

1. 0

- . 060

- .003

1.1

-120

- 911

PATA0015

[illegible]

1.2

-120

- .021

1.3

-285

- .031

DATA0016

[illegible]

1.4

- 360

-043

1.5

- .425

- 059

DATA0017

[illegible]

DATA001E

100

DATA0019

1. Introduction

DATA0020

150

-0.6

-260

- .036

-0.7

- . 320

-.042

DATA0021

[illegible]

-0.8

-390

-.050

-0.3

-.48v

- .057

DATA0022

[illegible]

-1.0

-545

- .067

-1.1

-620

- 077

DATA0023

[illegible]

10000029

[illegible]

DATA0025

[illegible]

DATA0025.

[illegible]

DATA0027

ADC 5001

DATE 0022

405 508.

DATA0029

ABC

0.8

.050

0. 0

0.9

.100

.010

DATA0030

[illegible]

1.0

.160

.012

1.1

.220

.027

DATA0031

[illegible]

1.2

.310

038

1.3

390

.050

DATA0032

[illegible]

0.0	.200	.400	.600	.800	1.000	DATE: 01/31
1/J(I)	E(I)	EE(I)	1/J(I+1)	E(I+1)	EE(I+1)	
00000000	00000000	00000000	00000000	00000000	00000000	00
11111111	11111111	11111111	11111111	11111111	11111111	11
22222222	22222222	22222222	22222222	22222222	22222222	22
33333333	33333333	33333333	33333333	33333333	33333333	33
44444444	44444444	44444444	44444444	44444444	44444444	44
55555555	55555555	55555555	55555555	55555555	55555555	55
66666666	66666666	66666666	66666666	66666666	66666666	66
77777777	77777777	77777777	77777777	77777777	77777777	77
88888888	88888888	88888888	88888888	88888888	88888888	88
99999999	99999999	99999999	99999999	99999999	99999999	99

ADD 5001

Data Cards 34-49

FORMAT 2 (10x, 2F10.4)
ELEMENTS

1/J = Inverse non dimensional speed . Note that as J goes to 1.5 it is only necessary to carry 1/J to 1/1.5 \approx .7 for coverage of full range.

E = Modified Thrust coefficient (non-dimensional) = C_T

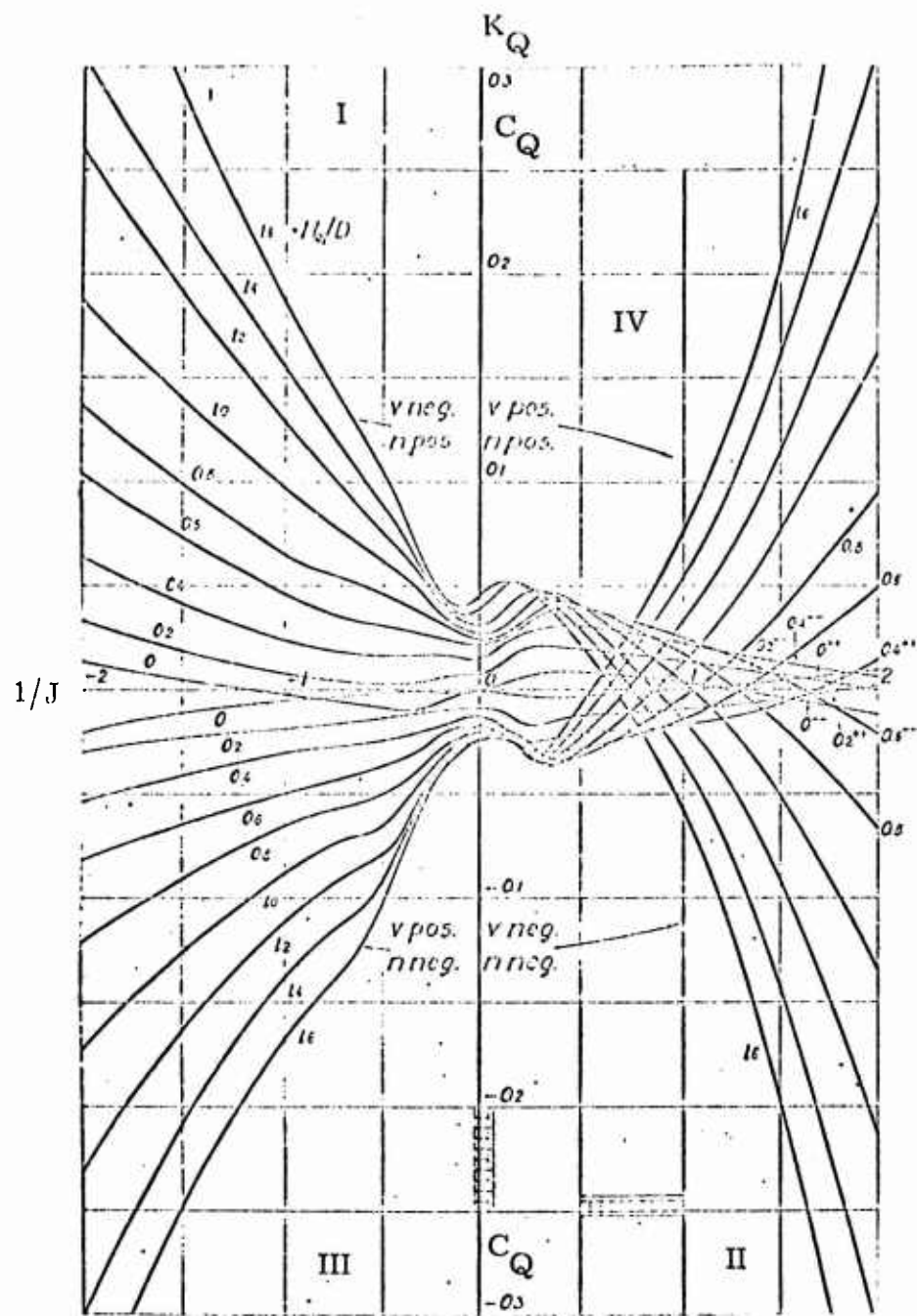
$$C_T = \frac{T}{\rho V_a^2 D^2}$$

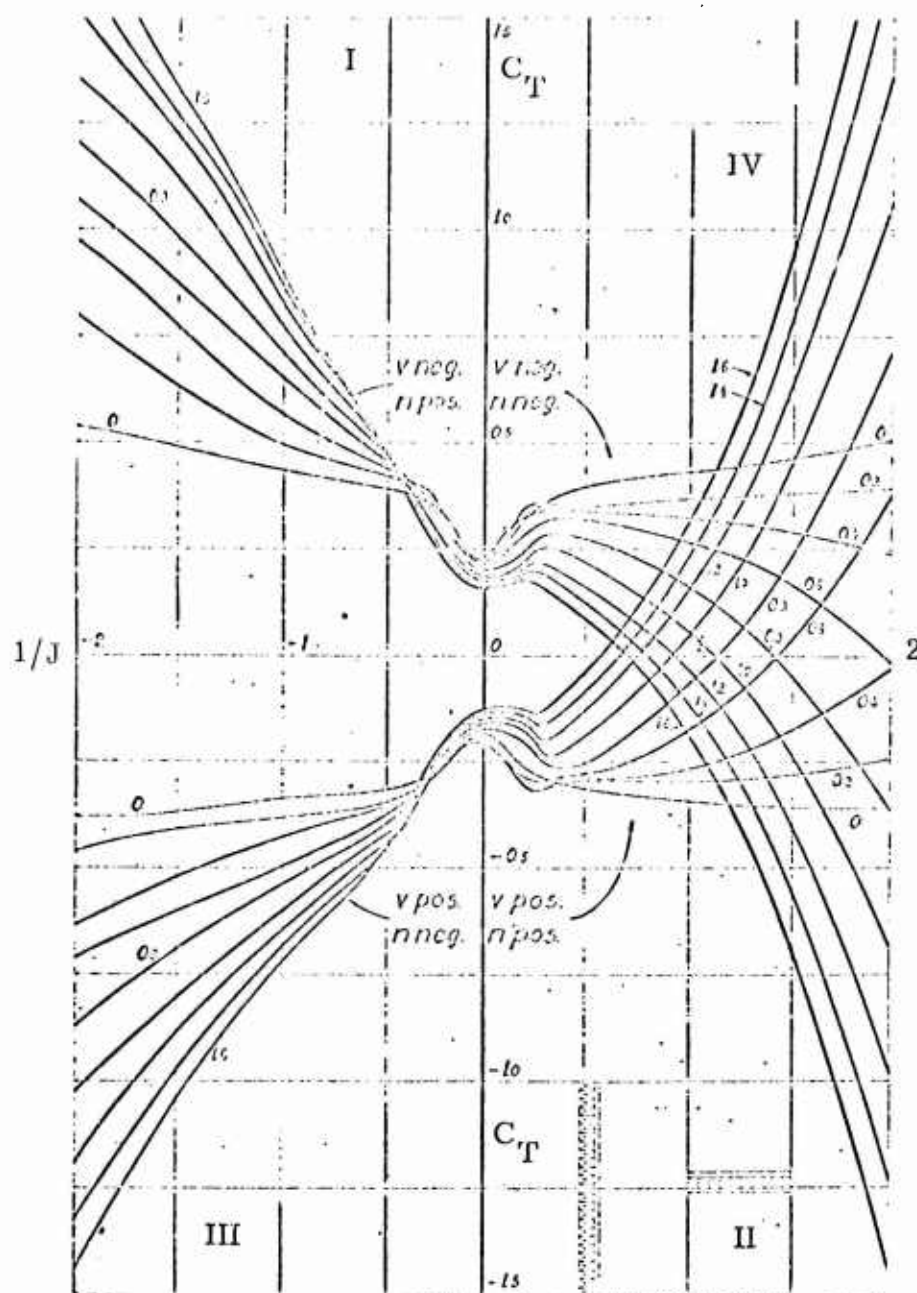
EE = Modified Torque coefficient (non-dimensional) = C_Q

$$C_Q = \frac{Q_p}{\rho V_a^2 D^3}$$

These elements are read from standard propeller curves (Miniovich or Nordstrom) in the order of quadrants I, II, III, IV as shown on the following figures. Note that the 1/J spacing must be as shown on the example cards as the 1/J value is not read, it is assumed by the program.

Note: Written description continues on page 42





DATA0035'

AUC 5051

DATA0036

ADC 5061

037001

406 5031

0.0

- 174 -

023

0.1

- 13

- 020

DATA0035

[illegible]

400 5051

0.2'

- 20

- 029

0.3

.. 25

- 030

DATE: 0979

[illegible]

ADC 5021

0.4

27

035

0.5

- 25 -

- 032

DATA002.C

[illegible]

40C 5001

DATE 0041

APC 6001

DATA0042

40C 6651

DATA0043

400 500:

DATA04:

DATA0045

DATA0040

-39-

19410541

400 500 1

DATA004

ADP 5691

DATA0049

600 5051

HULL DATA

DATA0050

[illegible]

Data Card 50

FORMAT 3F10.4

ELEMENTS

CAD = Added mass coefficient (non-dimensional)
Use 8%.

DEMTA = Ship displacement (long tons)

XKK .. Number of propellers.

DATA0051

[illegible]

FORMAT 10x, 3F10.4

ELEMENTS

VK = Speed of ship (knots)

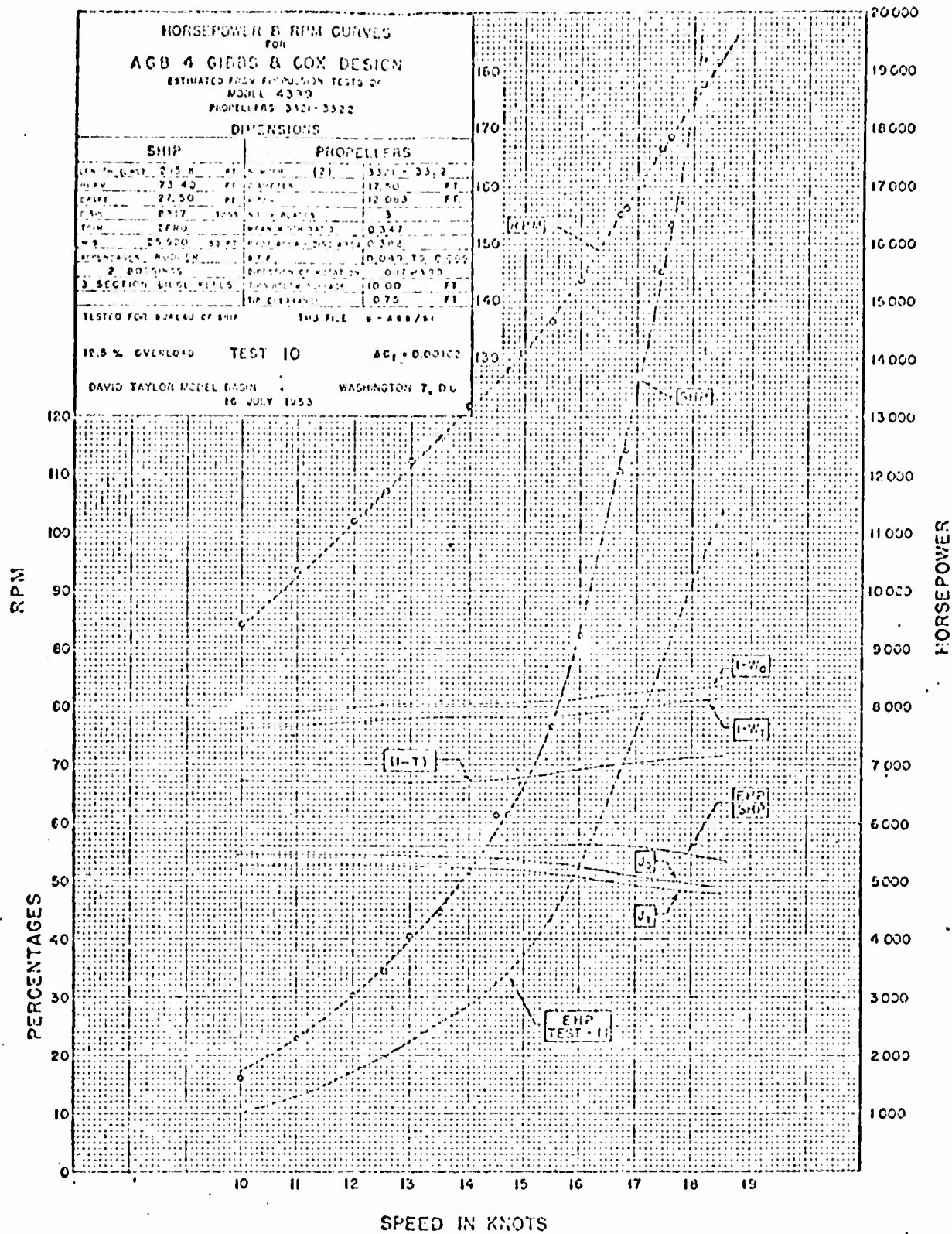
EHP = , Effective horsepower (horsepower)

WADED	Wake deduction $(1-W_T)$ (non dimensional)
0.00	0.00
0.05	0.05
0.10	0.10
0.15	0.15
0.20	0.20
0.25	0.25
0.30	0.30
0.35	0.35
0.40	0.40
0.45	0.45
0.50	0.50
0.55	0.55
0.60	0.60
0.65	0.65
0.70	0.70
0.75	0.75
0.80	0.80
0.85	0.85
0.90	0.90
0.95	0.95
1.00	1.00

THDED = Thrust deduction (1-T) (non-dimensional)

Each of the elements are read from the ship curves such as those following. VK spacing must be as given as it is assumed by the program, not read. Values for EHP, $1-w$, and $1-t$ not given on the curve can be found from straight line measurements back to origin for $1-t$ and $1-w$ and a square law for EHP.

Note: Written description continues on page 53.



1991-92-3

ABC 5021

Introduction

ADC 3051

DAY:0054

AUG 2001

08760053

ADC 5001

DATA0059

ADC 5081

DATA005?

40C 50E 1

1967110600

APC 5031

DATA0059

100 5051

DATA0000

400 500

DATA0051

ADC 6051

DATE 0022

101 5031

DATA00E3

400 500 600

DATE 0024

14.0	4127.0	894	764
------	--------	-----	-----

000000

15.0	5372.0	77,890	770
------	--------	--------	-----

DATA0066

[illegible]

10740067

17.0	9795.0	.887	.750
------	--------	------	------

DATA0068

[illegible]

18.0	15001.0	77.908	.727
------	---------	--------	------

19AT0069

[illegible]

ELECTRICAL DATA

[illegible]

4F10.6

-53-

$Z =$ number of armature force conductors

$s =$ armature speed in rpm

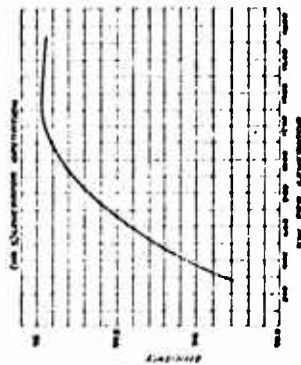
$$Z'_m = \frac{P_1 Z}{P_2 \cdot 60} \quad \text{and} \quad e_m = Z'_m \phi s$$

data for this is read from the technical manual or design sheets for the equipment.

RFM = Resistance of motor shunt field (ohms)

- 1950-1951 - 1952-1953
 1954-1955 - 1956-1957
 1958-1959 - 1960-1961
 1962-1963 - 1964-1965
 1966-1967 - 1968-1969
 1970-1971 - 1972-1973
 1974-1975 - 1976-1977
 1978-1979 - 1980-1981
 1982-1983 - 1984-1985
 1986-1987 - 1988-1989
 1990-1991 - 1992-1993
 1994-1995 - 1996-1997
 1998-1999 - 2000-2001
 2002-2003 - 2004-2005
 2006-2007 - 2008-2009
 2010-2011 - 2012-2013
 2014-2015 - 2016-2017
 2018-2019 - 2020-2021
 2022-2023 - 2024-2025
 2026-2027 - 2028-2029
 2030-2031 - 2032-2033
 2034-2035 - 2036-2037
 2038-2039 - 2040-2041
 2042-2043 - 2044-2045
 2046-2047 - 2048-2049
 2050-2051 - 2052-2053
 2054-2055 - 2056-2057
 2058-2059 - 2060-2061
 2062-2063 - 2064-2065
 2066-2067 - 2068-2069
 2070-2071 - 2072-2073
 2074-2075 - 2076-2077
 2078-2079 - 2080-2081
 2082-2083 - 2084-2085
 2086-2087 - 2088-2089
 2090-2091 - 2092-2093
 2094-2095 - 2096-2097
 2098-2099 - 2100-2101
 2102-2103 - 2104-2105
 2106-2107 - 2108-2109
 2110-2111 - 2112-2113
 2114-2115 - 2116-2117
 2118-2119 - 2120-2121
 2122-2123 - 2124-2125
 2126-2127 - 2128-2129
 2130-2131 - 2132-2133
 2134-2135 - 2136-2137
 2138-2139 - 2140-2141
 2142-2143 - 2144-2145
 2146-2147 - 2148-2149
 2150-2151 - 2152-2153
 2154-2155 - 2156-2157
 2158-2159 - 2160-2161
 2162-2163 - 2164-2165
 2166-2167 - 2168-2169
 2170-2171 - 2172-2173
 2174-2175 - 2176-2177
 2178-2179 - 2180-2181
 2182-2183 - 2184-2185
 2186-2187 - 2188-2189
 2190-2191 - 2192-2193
 2194-2195 - 2196-2197
 2198-2199 - 2200-2201
 2202-2203 - 2204-2205
 2206-2207 - 2208-2209
 2210-2211 - 2212-2213
 2214-2215 - 2216-2217
 2218-2219 - 2220-2221
 2222-2223 - 2224-2225
 2226-2227 - 2228-2229
 2230-2231 - 2232-2233
 2234-2235 - 2236-2237
 2238-2239 - 2240-2241
 2242-2243 - 2244-2245
 2246-2247 - 2248-2249
 2250-2251 - 2252-2253
 2254-2255 - 2256-2257
 2258-2259 - 2260-2261
 2262-2263 - 2264-2265
 2266-2267 - 2268-2269
 2270-2271 - 2272-2273
 2274-2275 - 2276-2277
 2278-2279 - 2280-2281
 2282-2283 - 2284-2285
 2286-2287 - 2288-2289
 2290-2291 - 2292-2293
 2294-2295 - 2296-2297
 2298-2299 - 2300-2301
 2302-2303 - 2304-2305
 2306-2307 - 2308-2309
 2310-2311 - 2312-2313
 2314-2315 - 2316-2317
 2318-2319 - 2320-2321
 2322-2323 - 2324-2325
 2326-2327 - 2328-2329
 2330-2331 - 2332-2333
 2334-2335 - 2336-2337
 2338-2339 - 2340-2341
 2342-2343 - 2344-2345
 2346-2347 - 2348-2349
 2350-2351 - 2352-2353
 2354-2355 - 2356-2357
 2358-2359 - 2360-2361
 2362-2363 - 2364-2365
 2366-2367 - 2368-2369
 2370-2371 - 2372-2373
 2374-2375 - 2376-2377
 2378-2379 - 2380-2381
 2382-2383 - 2384-2385
 2386-2387 - 2388-2389
 2390-2391 - 2392-2393
 2394-2395 - 2396-2397
 2398-2399 - 2400-2401
 2402-2403 - 2404-2405
 2406-2407 - 2408-2409
 2410-2411 - 2412-2413
 2414-2415 - 2416-2417
 2418-2419 - 2420-2421
 2422-2423 - 2424-2425
 2426-2427 - 2428-2429
 2430-2431 - 2432-2433
 2434-2435 - 2436-2437
 2438-2439 - 2440-2441
 2442-2443 - 2444-2445
 2446-2447 - 2448-2449
 2450-2451 - 2452-2453
 2454-2455 - 2456-2457
 2458-2459 - 2460-2461
 2462-2463 - 2464-2465
 2466-2467 - 2468-2469
 2470-2471 - 2472-2473
 2474-2475 - 2476-2477
 2478-2479 - 2480-2481
 2482-2483 - 2484-2485
 2486-2487 - 2488-2489
 2490-2491 - 2492-2493
 2494-2495 - 2496-2497
 2498-2499 - 2500-2501
 2502-2503 - 2504-2505
 2506-2507 - 2508-2509
 2510-2511 - 2512-2513
 2514-2515 - 2516-2517
 2518-2519 - 2520-2521
 2522-2523 - 2524-2525
 2526-2527 - 2528-2529
 2530-2531 - 2532-2533
 2534-2535 - 2536-2537
 2538-2539 - 2540-2541
 2542-2543 - 2544-2545
 2546-2547 - 2548-2549
 2550-2551 - 2552-2553
 2554-2555 - 2556-2557
 2558-2559 - 2560-2561
 2562-2563 - 2564-2565
 2566-2567 - 2568-2569
 2570-2571 - 2572-2573
 2574-2575 - 2576-2577
 2578-2579 - 2580-2581
 2582-2583 - 2584-2585
 2586-2587 - 2588-2589
 2590-2591 - 2592-2593
 2594-2595 - 2596-2597
 2598-2599 - 2600-2601
 2602-2603 - 2604-

- (a) 20 miles - 25 minutes.
- (b) Type - Two Bays - better to approach Main and secondary buildings.
- (c) Starting Depth - 1/2 inch per Bay.
- (d) Bay Area Indicated - 2,000 sq. ft.
- (e) Direction to Street - 1/2 inch from 15 meters - 1/2 inch from 15.



RG

TRNFG

ZGP

RFG

[illegible]

Data Card 72

FORMAT

4F10.6

ELEMENTS

Each of these elements is found from data on the equipment such as from the certification data shown for the GLACIER. Note that values are on a per shaft basis.

RG =

Resistance of generator armature circuit including series field and commutating windings.

TRNFG =

Turns of generator shunt field winding/pole.

$$ZGP' =$$
$$Z_g' = \text{flux conversion factor (volts/weber-rpm) from } P_1 \phi Z_s$$

$$e_g = \frac{P_1 \phi Z_s}{P_2 \cdot 60}$$

$$Z'_g = \frac{P_1 Z}{P_{260}} \quad \text{and} \quad e_g = Z'_g \phi_s$$

data for this is read from the technical manual or design sheets for the equipment.

RFG =

resistance of generator shunt field (ohms).

[illegible]

The other described branches will continue to Military Specialties, except that, in 1977, there will be no new appointments in the senior ranks of the Army, Navy, and Air Force. The Air Force will continue to have no appointments in the senior ranks of the Air Force, and the Navy will continue to have no appointments in the senior ranks of the Navy. The Air Force will continue to have no appointments in the senior ranks of the Air Force, and the Navy will continue to have no appointments in the senior ranks of the Navy.

[illegible]

- (a) 100/1000 Poles.
- (b) 1000/1000 Poles.
- (c) 1000/1000 Poles.
- (d) 1000/1000 Poles.
- (e) 1000/1000 Poles.

[illegible]

卷之四

1891	1892	1893	1894	1895	1896	1897	1898	1899	1900	1901	1902	1903	1904	1905	1906	1907	1908	1909	1910	1911	1912	1913	1914	1915	1916	1917	1918	1919	1920	1921	1922	1923	1924	1925	1926	1927	1928	1929	1930	1931	1932	1933	1934	1935	1936	1937	1938	1939	1940	1941	1942	1943	1944	1945	1946	1947	1948	1949	1950	1951	1952	1953	1954	1955	1956	1957	1958	1959	1960	1961	1962	1963	1964	1965	1966	1967	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299</
------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	--------

(b) **Source: United States**

A **B**

2. United States of America (2)

3. United Kingdom (2)

- (a) Type - *Monocot*
- (b) Internal - *Scoring cells are not found. Hard to hold in water. No chlorophyll. 10-15 µm.*
- (c) Intercells - *Parquet. Suberization from a separate outer dermal layer.*
- (d) Diameter - *17 µm*
- (e) Length - *15 µm*

- (1) Refer to "General Manager and Public Agent", Bureau of Investigation.
- (2) From file 100-364.
- (3) Commission - 12 days in Berlin, 2 days in Paris.
- (4) Indemnity - \$2,000 plus \$100.

[illegible]

(1) 1. The first of the two is the first of the two
 (2) 2. The second of the two is the second of the two
 (3) 3. The third of the two is the third of the two
 (4) 4. The fourth of the two is the fourth of the two
 (5) 5. The fifth of the two is the fifth of the two
 (6) 6. The sixth of the two is the sixth of the two
 (7) 7. The seventh of the two is the seventh of the two
 (8) 8. The eighth of the two is the eighth of the two
 (9) 9. The ninth of the two is the ninth of the two
 (10) 10. The tenth of the two is the tenth of the two
 (11) 11. The eleventh of the two is the eleventh of the two
 (12) 12. The twelfth of the two is the twelfth of the two
 (13) 13. The thirteenth of the two is the thirteenth of the two
 (14) 14. The fourteenth of the two is the fourteenth of the two
 (15) 15. The fifteenth of the two is the fifteenth of the two
 (16) 16. The sixteenth of the two is the sixteenth of the two
 (17) 17. The seventeenth of the two is the seventeenth of the two
 (18) 18. The eighteenth of the two is the eighteenth of the two
 (19) 19. The nineteenth of the two is the nineteenth of the two
 (20) 20. The twentieth of the two is the twentieth of the two
 (21) 21. The twenty-first of the two is the twenty-first of the two
 (22) 22. The twenty-second of the two is the twenty-second of the two
 (23) 23. The twenty-third of the two is the twenty-third of the two
 (24) 24. The twenty-fourth of the two is the twenty-fourth of the two
 (25) 25. The twenty-fifth of the two is the twenty-fifth of the two
 (26) 26. The twenty-sixth of the two is the twenty-sixth of the two
 (27) 27. The twenty-seventh of the two is the twenty-seventh of the two
 (28) 28. The twenty-eighth of the two is the twenty-eighth of the two
 (29) 29. The twenty-ninth of the two is the twenty-ninth of the two
 (30) 30. The thirtieth of the two is the thirtieth of the two
 (31) 31. The thirty-first of the two is the thirty-first of the two
 (32) 32. The thirty-second of the two is the thirty-second of the two
 (33) 33. The thirty-third of the two is the thirty-third of the two
 (34) 34. The thirty-fourth of the two is the thirty-fourth of the two
 (35) 35. The thirty-fifth of the two is the thirty-fifth of the two
 (36) 36. The thirty-sixth of the two is the thirty-sixth of the two
 (37) 37. The thirty-seventh of the two is the thirty-seventh of the two
 (38) 38. The thirty-eighth of the two is the thirty-eighth of the two
 (39) 39. The thirty-ninth of the two is the thirty-ninth of the two
 (40) 40. The fortieth of the two is the fortieth of the two
 (41) 41. The forty-first of the two is the forty-first of the two
 (42) 42. The forty-second of the two is the forty-second of the two
 (43) 43. The forty-third of the two is the forty-third of the two
 (44) 44. The forty-fourth of the two is the forty-fourth of the two
 (45) 45. The forty-fifth of the two is the forty-fifth of the two
 (46) 46. The forty-sixth of the two is the forty-sixth of the two
 (47) 47. The forty-seventh of the two is the forty-seventh of the two
 (48) 48. The forty-eighth of the two is the forty-eighth of the two
 (49) 49. The forty-ninth of the two is the forty-ninth of the two
 (50) 50. The fiftieth of the two is the fiftieth of the two
 (51) 51. The fifty-first of the two is the fifty-first of the two
 (52) 52. The fifty-second of the two is the fifty-second of the two
 (53) 53. The fifty-third of the two is the fifty-third of the two
 (54) 54. The fifty-fourth of the two is the fifty-fourth of the two
 (55) 55. The fifty-fifth of the two is the fifty-fifth of the two
 (56) 56. The fifty-sixth of the two is the fifty-sixth of the two
 (57) 57. The fifty-seventh of the two is the fifty-seventh of the two
 (58) 58. The fifty-eighth of the two is the fifty-eighth of the two
 (59) 59. The fifty-ninth of the two is the fifty-ninth of the two
 (60) 60. The sixtieth of the two is the sixtieth of the two
 (61) 61. The sixty-first of the two is the sixty-first of the two
 (62) 62. The sixty-second of the two is the sixty-second of the two
 (63) 63. The sixty-third of the two is the sixty-third of the two
 (64) 64. The sixty-fourth of the two is the sixty-fourth of the two
 (65) 65. The sixty-fifth of the two is the sixty-fifth of the two
 (66) 66. The sixty-sixth of the two is the sixty-sixth of the two
 (67) 67. The sixty-seventh of the two is the sixty-seventh of the two
 (68) 68. The sixty-eighth of the two is the sixty-eighth of the two
 (69) 69. The sixty-ninth of the two is the sixty-ninth of the two
 (70) 70. The seventieth of the two is the seventieth of the two
 (71) 71. The seventy-first of the two is the seventy-first of the two
 (72) 72. The seventy-second of the two is the seventy-second of the two
 (73) 73. The seventy-third of the two is the seventy-third of the two
 (74) 74. The seventy-fourth of the two is the seventy-fourth of the two
 (75) 75. The seventy-fifth of the two is the seventy-fifth of the two
 (76) 76. The seventy-sixth of the two is the seventy-sixth of the two
 (77) 77. The seventy-seventh of the two is the seventy-seventh of the two
 (78) 78. The seventy-eighth of the two is the seventy-eighth of the two
 (79) 79. The seventy-ninth of the two is the seventy-ninth of the two
 (80) 80. The eightieth of the two is the eightieth of the two
 (81) 81. The eighty-first of the two is the eighty-first of the two
 (82) 82. The eighty-second of the two is the eighty-second of the two
 (83) 83. The eighty-third of the two is the eighty-third of the two
 (84) 84. The eighty-fourth of the two is the eighty-fourth of the two
 (85) 85. The eighty-fifth of the two is the eighty-fifth of the two
 (86) 86. The eighty-sixth of the two is the eighty-sixth of the two
 (87) 87. The eighty-seventh of the two is the eighty-seventh of the two
 (88) 88. The eighty-eighth of the two is the eighty-eighth of the two
 (89) 89. The eighty-ninth of the two is the eighty-ninth of the two
 (90) 90. The ninetieth of the two is the ninetieth of the two
 (91) 91. The ninety-first of the two is the ninety-first of the two
 (92) 92. The ninety-second of the two is the ninety-second of the two
 (93) 93. The ninety-third of the two is the ninety-third of the two
 (94) 94. The ninety-fourth of the two is the ninety-fourth of the two
 (95) 95. The ninety-fifth of the two is the ninety-fifth of the two
 (96) 96. The ninety-sixth of the two is the ninety-sixth of the two
 (97) 97. The ninety-seventh of the two is the ninety-seventh of the two
 (98) 98. The ninety-eighth of the two is the ninety-eighth of the two
 (99) 99. The ninety-ninth of the two is the ninety-ninth of the two
 (100) 100. The hundredth of the two is the hundredth of the two

- (a) Diameter = 300 microns, length of wire = 10 microns
- (b) Heating = electric to operating regime at 1000°C; duration of heating 10 min
- (c) Initial of heating temperature = 200°C; duration of heating = 10 min; time held at 1000°C = 10 min; duration of cooling = 10 min
- (d) Measurements are taken together with the picture taken in the picture.
- (e) Power of 100 W = 300.
- (f) Conductivity per 1000 = 6.
- (g) Measurements at 170°C = 0.0001 cm.
- (h) Nature of particles: point = 10.

- (a) Diameter - 5 1/2" x 5" Section
- (b) Type - 6"
- (c) Working Depth - 5" and 1/2" and 1/4"
- (d) Bar Size Diameter - 1 1/2" Section

[illegible][illegible]

Graph showing Percent Polymer (Y-axis, 0 to 100) versus Time (min) (X-axis, 0 to 100). The curve represents the polymerization of K-4000/100 at 100°C. The rate of polymerization increases rapidly initially, reaching a maximum around 10 minutes, and then gradually decreases.

D4535000

ADC 5084

... ..

... ..

... ..

field current (amps)

offset of motor controller curve (volts)(see following figure)

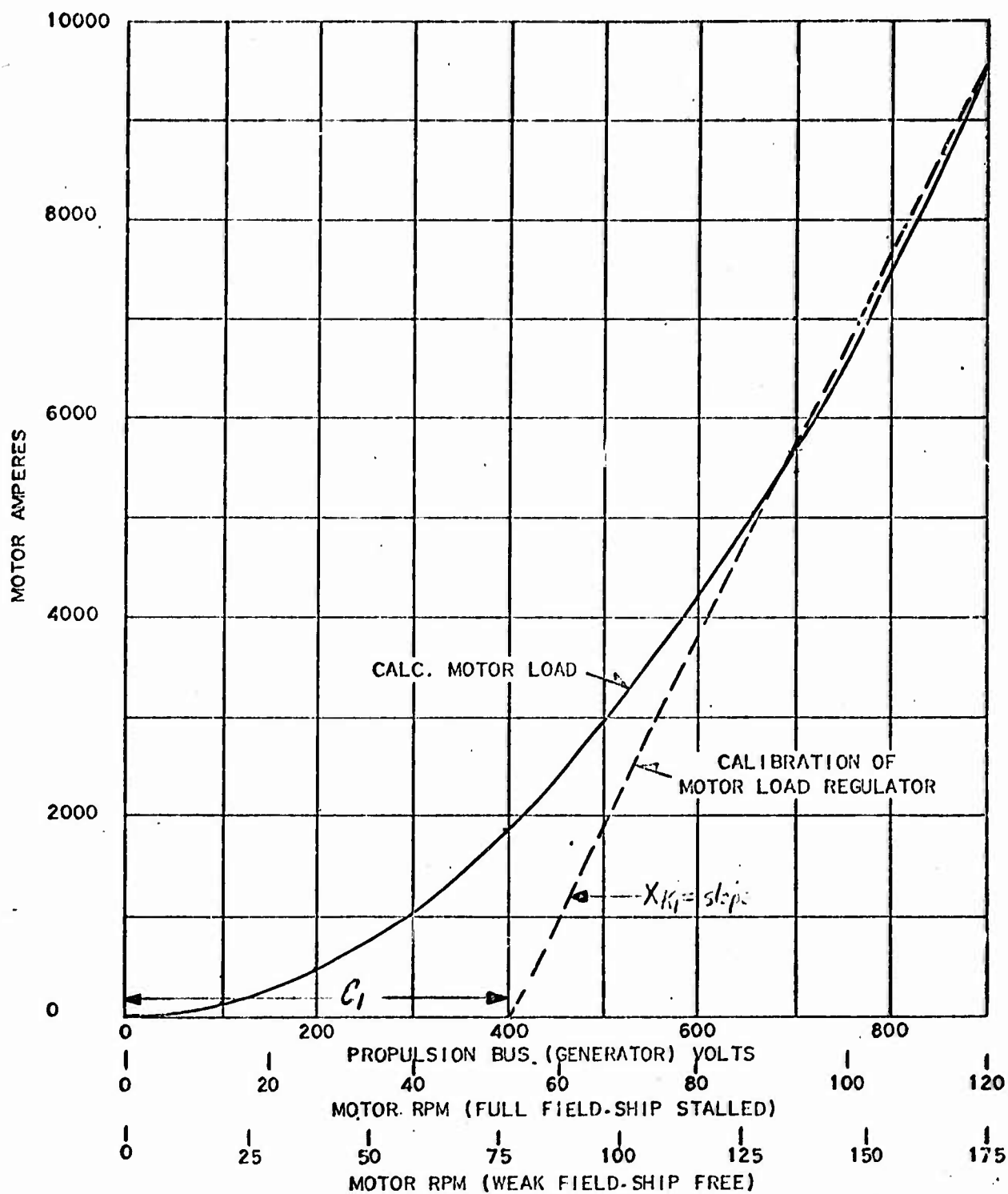
slope of motor controller curve (amps/volt) (see following figure)

shaft frictional coefficient (foot-lb-sec²)

$$Q_{\text{rost}} = C_{\text{SHFR}} (n_p) (|n_p|)$$

generator series field turns/pole

resistance (ohms) of the variable resistor in the generator field circuit.



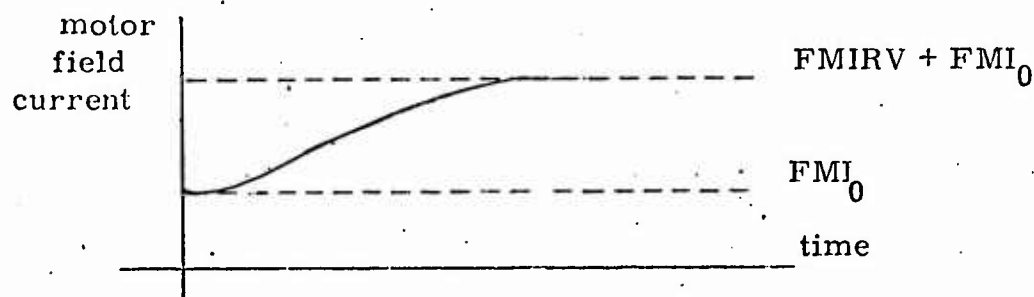
• Approximate Motor Load Currents

[illegible]

3F10.4

ELEMENTS

On the GLACIER, the motor field is plugged in the reverse direction during a crash back approximately as follows:



FMIRV = value in amperes per above sketch

TAUA, TAUB = exponent values to fit the following equation to the above sketch.

$$FMI = FMI_0 + \frac{FMIRV}{TAUA - TAUB} (TAUA e^{-t} / TAUA^t -$$

$$TAUB \cdot e^{-t/TAUB})$$

[illegible]

Data card 75:

FORMAT F10.4, F10.7

ELEMENTS

XGEN = number of diesel generator sets

X_{LA} = total inductance of armature circuit for one shaft
(henries)

CF₃MA(I)

APC 5091

Figure 1 consists of two scatter plots. The left plot shows a positive correlation between the number of children and the number of mothers, with a regression line indicating a positive slope. The right plot shows a negative correlation between the number of children and the number of mothers, with a regression line indicating a negative slope.

2(10x, F10.4)

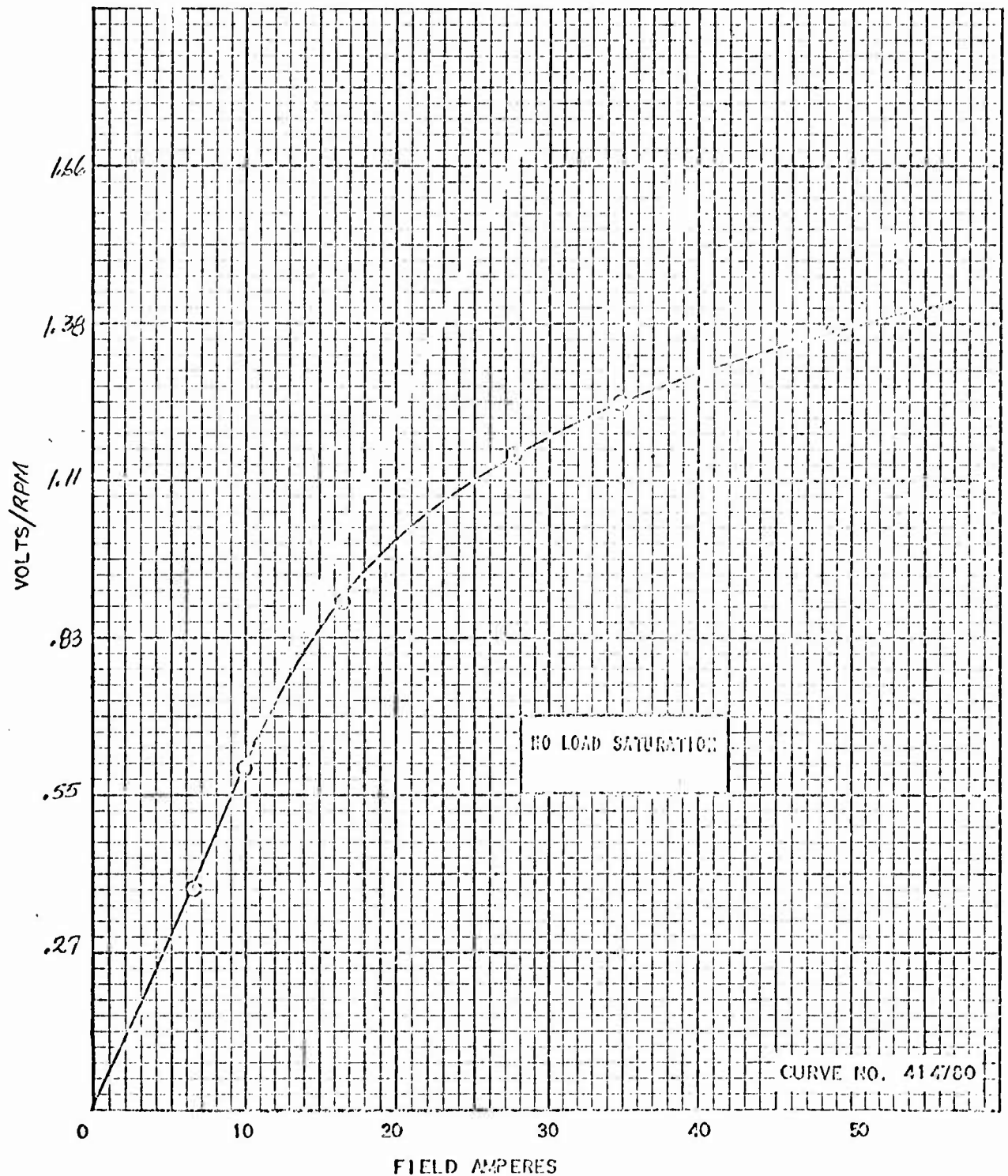
re derived from

constant times the generator field magnetic flux
(volts/rpm)

constant times the motor field magnetic flux
(volts/rmp)

description conti

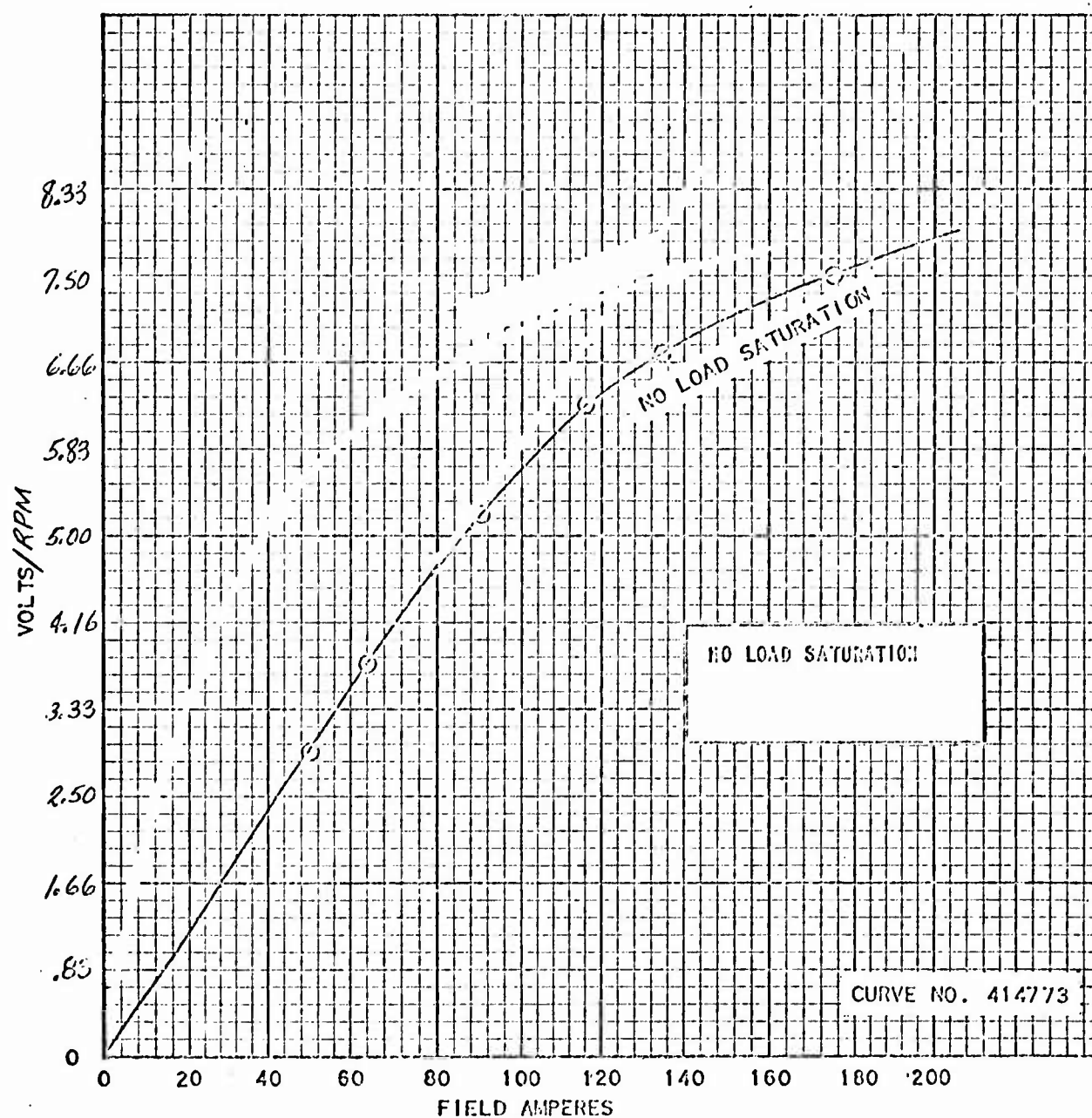
FRAME NO. CC-219.5 TYPE QH D.C. GENERATOR
 1340 KW - 837 VOLTS - 1600 AMP
 6 POLES 720 RPM
 U.S. NAVY CONTRACT NO. 2050



Saturation Curves of Propulsion Generator

ELECTRIC PROPULSION EQUIPMENT

FRAME NO. LH-429.7 - TYPE QV D.C. MOTOR
8450/10500 H.P. - 837/900 VOLTS - 8000/9300 AMPS
24 POLES - 112/175 RPM
U.S. NAVY CONTRACT NOs 2984



Propulsion Motor Saturation Curve

DATA0078

4427

DATA0079

THE 3000

DATA0080

1006 2421

DATA0034

144 80000

DATA0085

13V 5.00V

DATA0026

1473 3009

6.92

[illegible]

7.024

[illegible]

7.104

[illegible]

DIESEL DATA AND SYSTEM INERTIAS

ALP1 ALP2 ALP3 ALPD XK6 QBRG CWM

[illegible]

Data Card 90:

FORMAT

7F10.4

ELEMENTS

ALP1 = α_1 buffer spring scale *

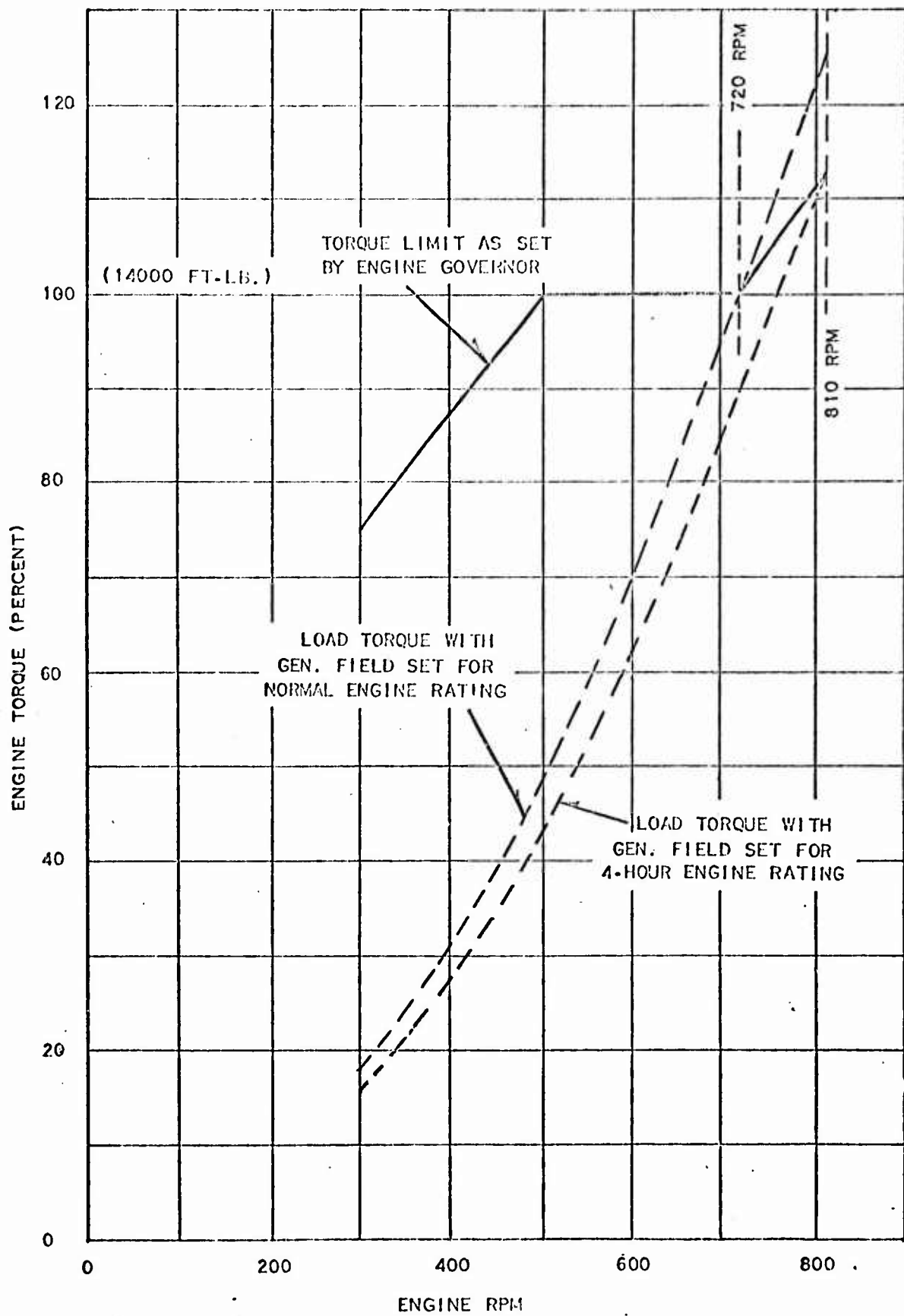
$$ALP_2 = \alpha_2 \text{ governor to engine constant}^*$$
$$ALP_3 = \alpha_3 \text{ needle value adjustment*}$$

ALPD = α_D = speed droop coefficient*

$$XK6 = \frac{\text{rated diesel load}}{\text{rated diesel stroke}} \left(\frac{\text{foot-lbs}}{\text{inch}} \right)$$

The rated diesel load is read from curves such as the attached. The rated diesel stroke is obtained from the manufacturer.

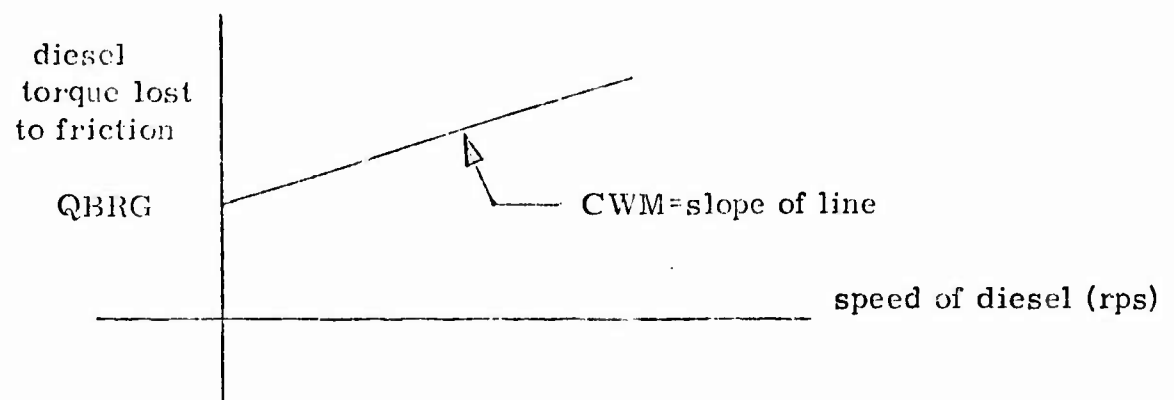
* See reference 1 for discussion of diesel governor data.



Engine Speed - Torque Characteristics

QBRG (foot-lbs) = see sketch below

CWM (foot-lb-sec) = see sketch below



0. 30930.

0.

97.2

302.

DATE0001

XJPA

XJM

XJSH

XJD

XJG

```

00000000 00000000 00000000 90000000 00000000 00000000 00000000 00000000 00000000 00000000
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111
22222222 22222222 22222222 22222222 22222222 22222222 22222222 22222222 22222222 22222222 22222222 22222222 22222222 22222222 22222222 22222222 22222222 22222222 22222222 22222222
33333333 3333 33 3 33333333 33333333 33333333 33333333 3 33333333 33333333 33333333 33333333 33333333 33333333 33333333 33333333 33333333 33333333 33333333
44444444 44444444 44444444 44444444 44444444 44444444 44444444 44444444 44444444 44444444 44444444 44444444 44444444 44444444 44444444 44444444 44444444 44444444 44444444 44444444
55555555 55555555 55555555 55555555 55555555 55555555 55555555 55555555 55555555 55555555 55555555 55555555 55555555 55555555 55555555 55555555 55555555 55555555 55555555 55555555
66666666 66666666 66666666 66666666 66666666 66666666 66666666 66666666 66666666 66666666 66666666 66666666 66666666 66666666 66666666 66666666 66666666 66666666 66666666 66666666
77777777 77777777 77777777 77777777 77777777 77777777 77777777 77777777 77777777 77777777 77777777 77777777 77777777 77777777 77777777 77777777 77777777 77777777 77777777 77777777
88888888 88888888 88888888 88888888 88888888 88888888 88888888 88888888 88888888 88888888 88888888 88888888 88888888 88888888 88888888 88888888 88888888 88888888 88888888 88888888
99999999 99999999 99999999 99999999 99999999 99999999 99999999 99999999 99999999 99999999 99999999 99999999 99999999 99999999 99999999 99999999 99999999 99999999 99999999 99999999
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
ADC 5001

```

Data Card 91

FORMAT

(7F16.4)

ELEMENTS

XJPA =

Polar moment of inertia of propeller in air
(ft-lb-sec²).

If given a value of zero on data card it
will be calculated as per subroutine POLM I.
Otherwise, given value will be used.

XJM =

Polar moment of inertia of one motor =

$$\frac{wk^2}{g} \quad (\text{ft} - \text{lb} - \text{sec}^2)$$

where g = value of gravity

wk^2 = given inertia factor for motor

XJSH =

Propeller shaft inertia (ft-lb-sec²). Usually
negligible.

XJD =

Polar moment of inertia of one diesel (ft-lb-sec²).
from manufacturers data.

XJG =

Polar moment of inertia of one generator =

$$\frac{wk^2}{g} \quad (\text{ft} - \text{lb} - \text{sec}^2)$$

ICE VALUES

[illegible]

Data Cards 92-96

FORMAT

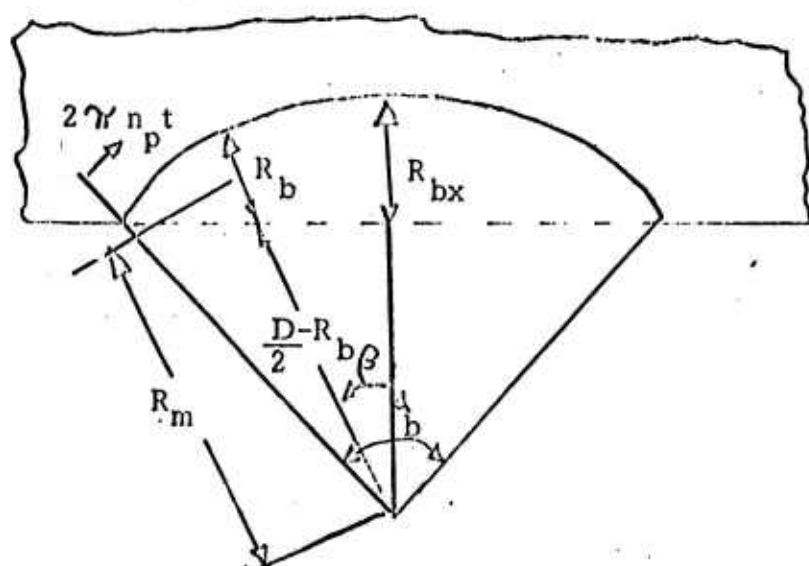
5F10.3

ELEMENTS

RBA =

$$\frac{R_{bx}}{D}$$

= depth of cut factor for propeller encountering ice. Values are non dimensional for each successive propeller blade. See following figure.



14316023

APC 5001

DATA0034

40C 5031

DATA0092

APC 5001

[illegible]

INITIAL VALUES AND SCALE FACTORS

	.0005	.005	.005	.0125	.0125	.0005	.125	DATA0001
	SY1	SY2	SY3	SY4	SY5	SY6	SY7	
000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	0 000
11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111	1 1111
22222222	22222222	22222222	22222222	22222222	22222222	22222222	22222222	2 2222
33333333	33333333	33333333	33333333	33333333	33333333	33333333	33333333	3 3333
44444444	44444444	44444444	44444444	44444444	44444444	44444444	44444444	4 4444
55555555	55555555	55555555	55555555	55555555	55555555	55555555	55555555	5 5555
66666666	66666666	66666666	66666666	66666666	66666666	66666666	66666666	6 6666
77777777	77777777	77777777	77777777	77777777	77777777	77777777	77777777	7 7777
88888888	88888888	88888888	88888888	88888888	88888888	88888888	88888888	8 8888
99999999	99999999	99999999	99999999	99999999	99999999	99999999	99999999	9 9999

ADC 5001

Data Card 98

FORMAT

7F10.7

ELEMENTS

Each of these elements are plotter scale factors - the actual number of inches along the Y-axis that corresponds to one (1) unit of the particular variable in question. The description of each plotter scale factor is as follows:

- SY1 = armature circuit current (amps)
- SY2 = bus volts
- SY3 = diesel speed (RPM)
- SY4 = motor field current (amps)
- SY5 = propeller speed (RPM)
- SY6 = power (KW)
- SY7 = vessel speed (KTS)

SY8	SY9	SY10	SY11	SY12	SX
-----	-----	------	------	------	----

[illegible]

Data Card 99,

FORMAT

6F10.7

ELEMENTS

Each of these elements are plotter scale factors - the actual number of inches along the axis in question (Y-axis for SY8, SY9, SY10, SY11 and SY12; X-axis for SX) that corresponds to one (1) unit of the variable in question. The description of each plotter scale factor is as follows:

SY8= propeller torque (ft-lbs)

SY9 = propeller thrust (lbs)

SY10 = distance (feet)

SY11 = Ice torque (ft-lbs)

SY12 = (not used)

SX = Time (seconds)

[illegible]

FORMAT 7F10.4

T = time program begins

THEIC = initial position of bridge controller in degrees between + 165 and -165.

XNPMD = initial rate of change of diesel speed (normally zero) (rev/sec²)

Y = initial rate of change of diesel rack position (rack velocity) (in/sec)

Z = initial rack position (inches) corresponding to initial ship horsepower for maneuver.

RV = reserve state variables

TEMP = reserve state variables.

U. U

DATN0102

A1

AIDC

[illegible]

Data Card 102

FORMAT

2F10.4

ELEMENTS

AI = initial armature current (amps)

AIDO = initial armature current derivative (amps/sec)

[illegible]

DATA0104

[illegible]

FORMAT

215, 4F10.4

ICNTR = variable defining type of maneuver (see subroutine CNTRL)

NON = program prints output every NON times DT seconds

DT = time step (seconds) for integration

TIMSC = time increment (seconds) for writing solution on disc.
TIMSC \geq DT and if TIMSC $>$ DT then TIMSC = Integer times DT

TICE = time (seconds) when start hitting ice

TEND = time (seconds) when program ends.

SAMPLE RUN

LISTING OF INPUT DECK


```

// JOB
// XEQ GENIS 1
#LOCALEXEC,CTRL,HEUN,FINIS
4
      0.0      .55      .8
    -0.2      .330     .035
    -0.4      .340     .034
    -0.6      .275     .032
    -0.8      .365     .040
    -1.0      .550     .050
    -1.2      .695     .064
    -1.4      .910     .081
    -1.6      1.160     .107
    -1.8      .320     .035
    -2.0      .280     .033
    -2.2      .220     .027
    -2.4      .140     .018
    -2.6      .050     .003
    -2.8      .060     .003
    -3.0      .150     .021
    -3.2      .350     .043
    -3.4      .220     .034
    -3.6      .170     .034
    -3.8      .260     .036
    -4.0      .390     .050
    -4.2      .545     .067
    -4.4      .730     .085
    -4.6      .930     .116
    -4.8      .820     .094
    -5.0      .190     .030
    -5.2      .130     .022
    -5.4      .070     .011
    -5.6      .050     .009
    -5.8      .100     .019
    -6.0      .310     .038
    -6.2      .450     .054
    -6.4      .200     .025
    -6.6      .250     .032
    -6.8      .410     .043
    -7.0      .560     .050
    -7.2      .175     .021
    -7.4      .20     .021
    -7.6      .27     .035
    -7.8      .22     .023
    -8.0      .175     .021
    -8.2      .230     .029
    -8.4      .330     .045

```

```

16.0      .125      1.082
    -0.1      .335      .035
    -0.3      .320      .033
    -0.5      .300      .033
    -0.7      .460      .045
    -0.9      .645      .057
    -1.1      .905      .072
    -1.3      1.0     .093
    -1.5      1.19     .119
    -1.7      .200     .034
    -1.9      .250     .030
    -2.1      .165     .024
    -2.3      .090     .015
    -2.5      0.0     .002
    -2.7      .120     .011
    -2.9      .205     .031
    -3.1      .425     .053
    -3.3      .220     .035
    -3.5      .180     .031
    -3.7      .220     .030
    -3.9      .350     .042
    -4.1      .430     .057
    -4.3      .520     .077
    -4.5      .830     .100
    -4.7      .100     .025
    -4.9      .210     .032
    -5.1      .170     .026
    -5.3      .100     .017
    -5.5      0.0     .007
    -5.7      .100     .010
    -5.9      .220     .027
    -6.1      .280     .050
    -6.3      .530     .080
    -6.5      .220     .028
    -6.7      .345     .037
    -6.9      .470     .046
    -7.1      .600     .055
    -7.3      .18     .020
    -7.5      .25     .030
    -7.7      .25     .033
    -7.9      .19     .022
    -8.1      .190     .024
    -8.3      .290     .038
    -8.5      .395     .050

```

5.12

```

DATA0001
DATA0002
DATA0003
DATA0004
DATA0005
DATA0006
DATA0007
DATA0008
DATA0009
DATA0010
DATA0011
DATA0012
DATA0013
DATA0014
DATA0015
DATA0016
DATA0017
DATA0018
DATA0019
DATA0020
DATA0021
DATA0022
DATA0023
DATA0024
DATA0025
DATA0026
DATA0027
DATA0028
DATA0029
DATA0030
DATA0031
DATA0032
DATA0033
DATA0034
DATA0035
DATA0036
DATA0037
DATA0038
DATA0039
DATA0040
DATA0041
DATA0042
DATA0043
DATA0044

```


.00001	.00001	.001	.0000010	.5	.625	DATA0099
3.0	165.0	0.0	0.0	0.3970	0.0	DATA0100
34.3854	13.3333	0.0	2.5387	29.3224	0.0	DATA0101
5997.1924	0.0	9.0	0.0	0.0	0.0	DATA0102
2	3.0	.04	300.	300.		DATA0103
5	.08					DATA0104

LISTING OF OUTPUT ON CONSOLE TYPEWRITER

FOR SAMPLE RUN

```
// XEQ GENIS 1
*LOCALEXEC,CNTRL,HEUN,FINIS
TYPE IN CONCLUSION OF PHRASE'SIMULATION OF'USING 30 CHARACTERS OR LESS
TEST SHIP FOR DOCUMENTATION' - - CRASH REVERSAL OF A 12000 TON SHIP
TURN ON APPROPRIATE SWITCHES
ICE DATA- 10
PROP DATA- 11
HULL DATA- 12
ELECTRICAL DATA- 13
GOVERNOR AND BRIDGE CONTROLLER DATA- 14
INERTIAS- 15
```

-93-

```
TURN ON SELECTOR SWITCHES AND PRESS START
TURN ON SWITCH --
1 - TO PUNCH NEW INITIAL CONDITION CARDS
2 - TO PLOT DATA
3 - TO CONTINUE SOLUTION OF CURRENT PROBLEM - SWITCH OFF STARTS NEW PROBLEM
4 - TO SUPPRESS FUTURE TYPING OF THESE INSTRUCTIONS
5 - TO CHANGE TIME STEP
6 - TO CHANGE PRINT INTERVAL
7 - TO CHANGE XLA
```

```
TURN ON SELECTOR SWITCHES AND PRESS START
AI--1,EB--2,XNPII--3,FHI--4,XNP--5,XRU--6,CVK--7,QPI--8,XPROI--9,CXS--10,QI--11,TEMP--12
```

```
WHICH VARIABLES TO BE PLOTTED (10 ONLY) -
( ) ( ) ( ) ( ) ( ) ( ) ( ) ( )
1 2 3 4 5 6 7 8 9 10
DO YOU WANT A GRID (0 - NO,1 - YES)
( )
```

```
1
SIMULATION OF TEST SHIP FOR DOCUMENTATION
CRASH REVERSAL OF A 12000 TON SHIP
```


LISTING OF OUTPUT ON PRINTER
AND PLOTTER

UNITED STATES COAST GUARD

ICEBREAKER PROPULSION SYSTEM SIMULATOR

-95-

SIMULATION OF TEST SHIP FOR DOCUMENTATION -- CRASH REVERSAL OF A 12000 TON SHIP
SERIAL NUMBER--
1.

SELECTED INPUT DATA

INITIAL CONDITIONS

T 0.0000 165.0000 THEIC 0.0000 XNPMD Y 0.0000 0.0000 RV 0.0000 TEMP 0.0000
 FGI 34.3364 13.3333 XNPM FID 0.0000 2.5397 VS 0.0000 AI 0.0000 AIDC 0.0000
 34.3364 13.3333 0.0000 2.5397 29.3224 0.1845 0.0000

CONTROL DATA

ICNTR= 2 NON = 5 DT = 0.0800 TICE= 300.0000 TEND = 300.0000 TIMSC= 0.0800
 THETI= 0.0000 TAU1 = 3.0000 TAU2 = 9.0000
 SY1 = 0.0005000 SY2 = 0.0050000 SY3 = 0.0050000 SY4 = 0.0125000 SY5 = 0.0125000
 SY6 = 0.0005000 SY7 = 0.1250000 SY8 = 0.0000100 SY9 = 0.0000100 SY10 = 0.0010000
 SY11 = 0.0000010 SY12 = 0.5000001 SX = 0.6250001

PROP DATA

IZR= 4 DAR= 0.5500 EO= 0.1250 DCT= 5.1200
 PDR= 0.8000 D = 16.0000 EK= 1.0820
 0.2300 0.0250 0.2000 0.0250
 0.3350 0.0350 0.2200 0.0280
 0.3400 0.0340 0.2000 0.0320
 0.3200 0.0330 0.3450 0.0370
 0.2750 0.0320 0.4100 0.0430
 0.2000 0.0320 0.4700 0.0460
 0.3650 0.0400 0.5600 0.0500
 0.4500 0.0450 0.6000 0.0550
 0.3500 0.0500 -0.01750 -0.0210
 0.6450 0.0570 -0.01800 -0.0200
 0.5050 0.0640 -0.02000 -0.0210
 0.9050 0.0720 -0.02500 -0.0200
 0.9100 0.0810 -0.02700 -0.0350
 1.0000 0.0930 -0.02500 -0.0330
 1.1600 0.1070 -0.02200 -0.0280
 1.1900 0.1180 -0.01900 -0.0220
 0.3200 0.0350 -0.01750 -0.0210
 0.3000 0.0340 -0.01900 -0.0240
 0.2800 0.0330 -0.02300 -0.0290
 0.2500 0.0320 -0.02900 -0.0380
 0.2200 0.0270 -0.03500 -0.0450
 0.1650 0.0240 -0.03950 -0.0500
 0.1400 0.0180 -0.04200 -0.0550
 0.0900 0.0160 -0.04500 -0.0570
 0.0500 0.0030 0.02000 0.0250
 0.0000 0.0020 0.02450 0.0260

-0.0600
 -0.1200
 -0.1800
 -0.2350
 -0.3600
 -0.4250
 -0.2200
 -0.2200
 -0.1900
 -0.1800
 -0.1700
 -0.2200
 -0.2600
 -0.3000
 -0.3900
 -0.4800
 -0.5450
 -0.6200
 -0.7300
 -0.8200
 -0.9200
 -1.0000
 -0.2200
 -0.2100
 -0.1900
 -0.1700
 -0.1300
 -0.1000
 -0.0700
 -0.0000
 0.0500
 0.1000
 0.1600
 0.2200
 0.3100
 0.3800
 0.4500
 0.5300

-0.0030
 -0.0110
 -0.0210
 -0.0310
 -0.0430
 -0.0530
 -0.0340
 -0.0350
 -0.0340
 -0.0310
 -0.0270
 -0.0200
 -0.0250
 -0.0420
 -0.0500
 -0.0570
 -0.0670
 -0.0770
 -0.0850
 -0.1000
 -0.1160
 -0.1250
 -0.0340
 -0.0320
 -0.0300
 -0.0260
 -0.0220
 -0.0170
 -0.0110
 -0.0070
 0.0000
 0.0100
 0.0180
 0.0270
 0.0380
 0.0500
 0.0640
 0.0800

0.2500
 0.2750
 0.2800
 0.2700
 0.2550
 0.2300

0.0300
 0.0370
 0.0430
 0.0420
 0.0360
 0.0330

HULL DATA

CAD= 0.0800 DELTA= 12000.0 XXX=

VK	EHP
0	0.0000
1	9.5000
2	37.8000
3	84.9000
4	151.2000
5	236.3000
6	340.2000
7	463.0000
8	686.0000
9	972.0000

3.

WADED

1.0000
0.9910
0.9830
0.9740
0.9650
0.9560
0.9480
0.9390
0.9300
0.9200

THDED

1.0000
0.9690
0.9380
0.9060
0.8750
0.8440
0.8130
0.7810
0.7500
0.7420

10	1926.0002	0.9180	0.7400
11	1768.0002	0.9100	0.7400
12	2363.0004	0.9040	0.7500
13	3148.0004	0.9000	0.7600
14	4127.0009	0.8940	0.7660
15	5372.0009	0.8900	0.7700
16	7046.0009	0.8840	0.7670
17	9795.0019	0.8870	0.7500
18	15001.0019	0.9080	0.7270
19	21209.0039	0.9080	0.7270

ELECTRICAL DATA

RM = 0.001520 RG = 0.009800 TRNFG = 495.0 ZGP = 4.60
 REG = 1.850000 TRNFX = 2160.00 ZMP = 32.40 RFM = 1.490000
 FIMAX = 84.0000 C1 = 10000.0019 XK1 = 0.0000
 CSUER = 1296.0002 TAUB = 0.0000 FMIRV = 0.0000 TAUUA = 0.0000
 TRNSE = 3.0000 RX = 1.7500 XGEN = 3.0000 XLA = 0.0002 SPGEN = 1000.0001 SPMOT = 20.0000

CFMA
 0.0000 5.6400
 0.2300 5.6400
 0.4780 5.6400
 0.7220 5.6400
 0.9250 5.6400
 1.0590 5.6400
 1.1500 5.6400
 1.2200 5.6400
 1.2600 5.6400
 1.2940 5.6400
 1.3250 5.6400
 1.3450 5.6400
 1.3660 5.6400

GOVERNOR AND DIESEL DATA

ALP1 = 7.2000 ALP2 = 0.0632 ALP3 = 1.1000 ALPD = 0.0000
 XK6 = 26650.00 GRRG = 2380.0004 CWM = 44.4000

SYSTEM INERTIAS

XJPA = 13312.00 XJM = 39930.00 XJSH = 0.00
 XJD = 97.20 XJG = 303.00 XJPE = 2001.24

ICE DATA

BLOC1 = 1000.000 DBR12 = 5.000 BLOC2 = 30.000 DBR21 = 5.000
 RPA


```
CCR= 5340.007 DSH= 13320.001 CICE= 4000.000
CK1= 0.000 CK2= 0.000 CK3= 0.000
```


OUTPUT DATA

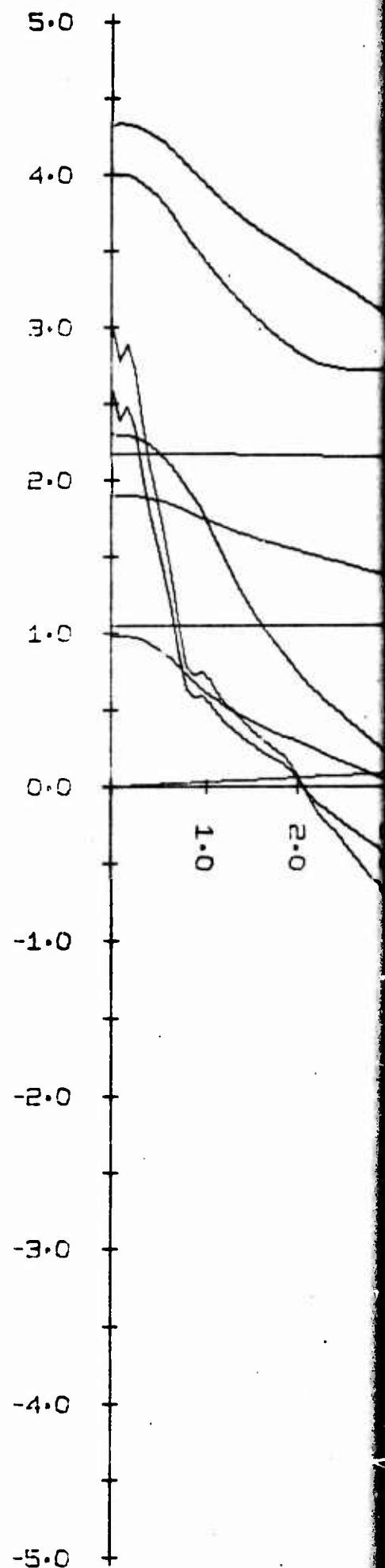
T(SEC)	RPVPM	CFG1	RPVPM	VK	QDE(FT-LB)	GG	ERI(V)	FMI	EC	QM(FT-LB)	AI(A)	QP	THRUST(T)	RESIST	HP/SH
0.000	799.9	34.3	152.3	17.3	46676.5	4678.4	864.8	84.0	859.0	238121.7	5997.1	229760.0	97.7	97.7	6905.0
0.399	790.9	31.0	150.6	17.3	19641.2	32765.8	854.9	84.0	849.7	167137.3	4209.4	222967.4	92.8	97.7	4794.9
0.799	717.9	26.8	143.6	17.3	8733.8	12495.6	811.7	84.0	810.3	61945.6	1560.1	195226.2	72.7	97.4	694.5
1.199	554.7	27.6	135.0	17.3	3523.5	8026.0	762.8	84.0	761.6	43006.7	1083.1	146571.0	52.1	96.6	1105.7
1.599	605.1	23.4	126.3	17.3	9493.3	5130.1	724.3	84.0	723.7	24573.5	610.8	104755.8	38.5	95.4	600.4
1.999	566.9	28.8	122.9	17.2	5336.6	1129.7	693.0	84.0	693.3	5163.4	130.0	74314.1	28.5	94.0	120.8
2.399	547.1	26.4	117.5	17.2	3344.6	4742.2	662.2	84.0	663.1	22159.7	599.1	50993.6	18.1	92.5	496.1
2.799	543.6	22.0	112.0	17.1	8346.9	914.8	630.3	84.0	632.0	46568.2	1172.8	29961.6	8.0	90.7	993.6
3.199	553.6	17.2	106.3	17.1	8369.0	12627.4	597.5	84.0	599.8	66594.5	1677.2	10677.9	-1.9	88.7	1343.4
3.599	560.6	13.7	100.7	17.0	8404.6	1428.5	565.3	84.0	568.2	80668.0	203.6	3953.7	-11.8	86.5	1547.4
3.999	580.3	10.3	95.1	17.0	8448.4	1563.4	533.1	84.0	536.6	99826.0	2514.1	17723.4	-20.5	84.1	1808.3
4.399	614.2	7.4	89.3	16.9	8503.7	1712.0	499.2	84.0	503.8	121886.8	3069.7	36178.0	-28.8	82.7	2073.0
4.799	644.4	4.0	83.0	16.8	8570.6	18407.4	462.8	84.0	469.5	151782.0	3822.6	60381.0	-36.4	81.5	2401.1
5.199	670.1	0.4	76.1	16.5	8647.8	2016.4	422.2	84.0	429.5	189496.4	4772.5	83769.4	-46.9	80.1	2747.5
5.599	716.2	-3.4	68.1	16.7	8730.1	2066.5	375.1	84.0	384.1	232219.5	5848.5	108759.1	-59.4	78.6	3011.3
5.999	752.2	-7.4	58.6	16.6	8809.9	1987.2	320.0	84.0	330.8	278201.1	7006.5	133124.5	-67.7	77.0	3107.5
6.399	782.3	-11.6	47.7	16.5	8877.9	1713.0	256.8	84.0	269.5	322805.3	8281.0	161443.4	-77.9	75.3	2992.1
6.799	802.0	-14.4	35.5	16.4	8920.4	1243.4	187.6	84.0	200.6	351877.2	8862.1	169389.4	-82.8	73.5	2383.8
7.199	809.2	-14.6	22.2	16.3	8923.2	615.6	113.9	84.0	125.3	324762.3	8179.2	121124.4	-68.7	71.7	1374.2
7.599	785.8	-14.3	8.4	16.3	8884.5	-187.5	37.3	84.0	47.6	287237.7	7234.1	97147.9	-55.4	70.2	461.9
7.999	753.4	-15.0	-3.8	16.2	8812.5	340.4	-31.7	84.0	-21.9	271641.8	6871.3	106818.4	-55.5	68.7	201.2
8.399	708.3	-16.0	-14.2	16.1	8587.9	773.1	-89.8	84.0	-80.4	260920.6	6571.3	127588.0	-64.6	67.2	708.3
8.799	653.2	-17.0	-22.2	16.0	8587.9	10712.4	-134.5	84.0	-125.3	253439.0	6382.9	156080.6	-77.7	65.5	1072.6
9.199	596.4	-17.7	-27.7	15.9	7642.0	13352.8	-164.9	84.0	-156.2	242410.1	6105.1	179584.3	-88.1	63.8	1278.6
9.599	539.4	-19.5	-31.6	15.8	10535.5	17437.1	-188.3	84.0	-178.5	253537.1	385.4	192575.3	-95.4	62.8	1525.4
9.999	535.9	-20.2	-35.5	15.3	2595.3	22400.0	-217.1	84.0	-206.0	287419.5	138.7	207509.3	-103.9	61.7	1999.2
10.399	552.9	-20.2	-42.4	15.7	3473.1	2736.0	-251.0	84.0	-239.1	310420.5	7818.0	220658.0	-111.7	60.6	2506.1
10.799	584.2	-20.2	-49.0	15.5	4200.0	32492.5	-289.1	84.0	-276.4	335922.0	8460.3	235525.1	-117.1	59.3	3134.9
11.199	611.3	-20.8	-55.5	15.4	4200.0	2453.6	-325.8	84.0	-313.2	326952.1	8486.2	244883.3	-121.8	58.1	3582.7
11.599	632.3	-27.3	-61.4	15.3	4200.0	36451.3	-359.5	84.0	-346.8	335747.1	8455.9	255532.4	-131.8	56.8	3930.9
11.999	647.3	-27.6	-66.8	15.2	4200.0	3716.7	-389.6	84.0	-377.2	331186.5	8341.0	253406.7	-136.0	55.4	4217.8
12.399	659.1	-27.3	-72.0	15.1	4200.0	38750.2	-418.3	84.0	-406.1	323775.6	8154.4	251980.2	-139.2	54.0	4439.0
12.799	665.8	-28.2	-76.4	15.0	4200.0	4022.7	-443.2	84.0	-431.0	323551.2	9148.7	263253.5	-137.5	52.6	4707.9
13.199	669.7	-28.6	-80.1	14.9	4200.0	41392.0	-464.1	84.0	-452.0	320316.3	8067.2	268368.3	-139.8	51.5	4888.1
13.599	669.4	-28.9	-83.5	14.8	4200.0	4215.7	-481.9	84.0	-470.0	316053.7	7959.9	271090.6	-141.9	50.5	5014.9
13.999	666.4	-29.1	-86.0	14.7	4200.0	4325.0	-496.9	84.0	-485.2	311010.0	7832.9	272452.6	-144.2	49.5	5095.0
14.399	661.8	-29.3	-89.3	14.5	4200.0	43790.0	-509.6	84.0	-498.1	305654.0	7698.0	272735.1	-145.8	48.4	5159.8
14.799	655.4	-29.5	-90.2	14.4	4200.0	4436.6	-520.1	84.0	-508.9	300120.0	7558.6	272170.1	-146.9	47.4	5155.5
15.199	647.6	-29.7	-91.7	14.3	4200.0	4496.0	-528.6	84.0	-517.4	295252.8	7436.0	273077.0	-146.2	46.3	5153.0
15.599	632.3	-30.0	-92.8	14.2	4200.0	4516.2	-534.8	84.0	-523.8	290689.6	7321.1	273836.3	-144.6	45.3	5141.0
15.999	627.8	-30.1	-93.0	14.1	4200.0	4523.7	-539.1	84.0	-528.3	285961.8	7202.0	273664.8	-143.2	44.2	5100.3
16.399	613.5	-30.2	-94.1	14.0	4200.0	4563.5	-541.7	84.0	-531.1	281130.3	700.3	272725.5	-141.5	43.1	5040.5
16.799	614.5	-30.1	-94.4	13.9	4200.0	4564.2	-542.9	84.0	-532.4	275398.0	6948.5	271208.2	-139.7	42.3	4959.4
17.199	592.5	-30.8	-94.4	13.8	4200.0	45787.2	-542.7	84.0	-532.4	270842.1	6821.2	269117.5	-137.9	41.5	4868.5
17.599	580.5	-31.0	-94.2	13.7	4200.0	4578.0	-541.4	84.0	-531.4	265861.4	6695.8	266577.0	-136.0	40.7	4769.5
17.999	563.8	-31.2	-93.8	13.6	4200.0	4550.2	-539.3	84.0	-529.5	261116.8	6576.3	263706.2	-134.1	40.0	4667.6

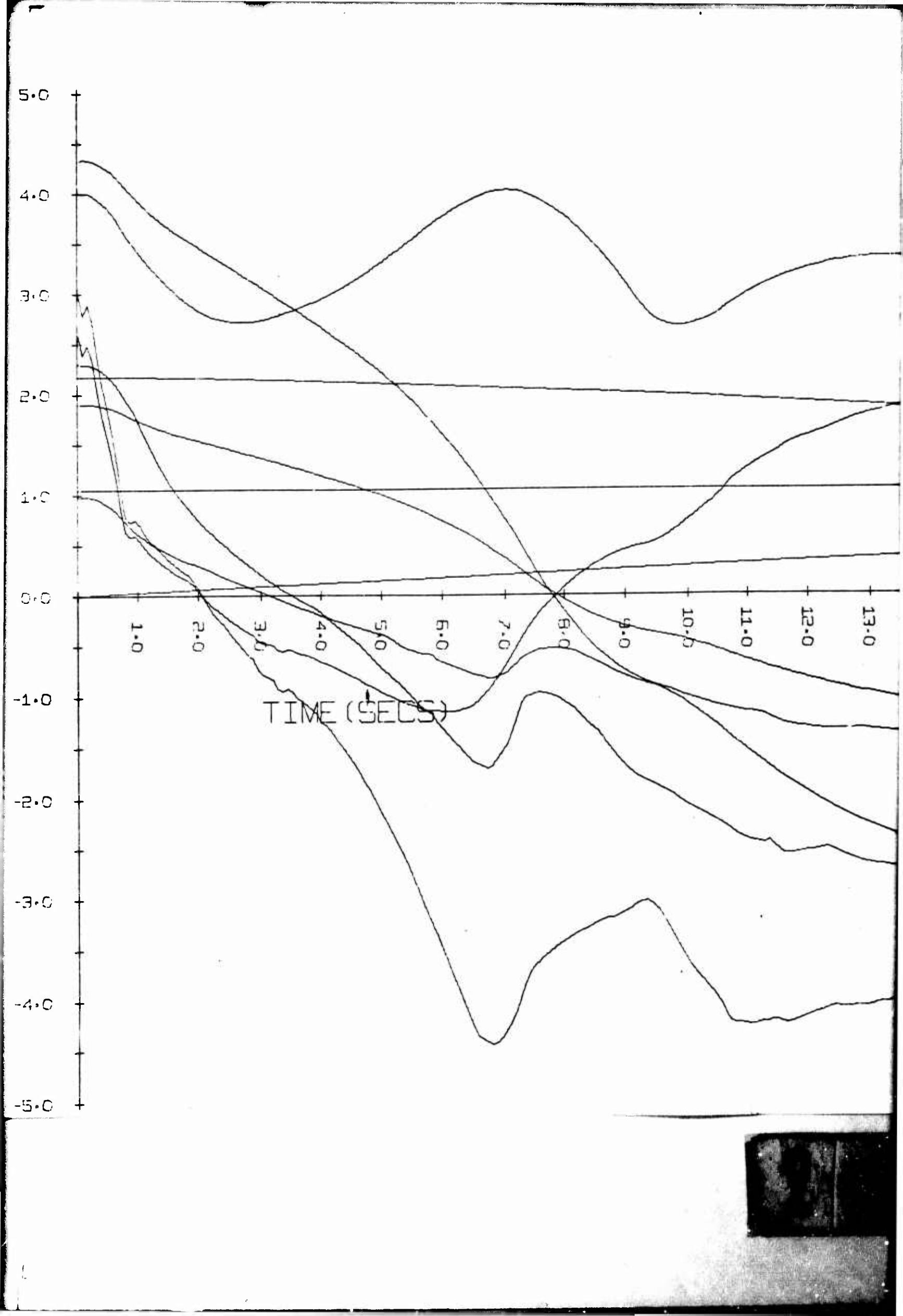
XS(1 IN°=1000°0 FEET)
 TP(1 IN°=100000°0 LBS)
 OP(1 IN°=100000°0 FT-LBS)
 VK(1 IN°= 8°0 KTS°)
 KW(1 IN°=2000°0 KW)
 NP(1 IN°= 80°0 RPM)
 FMI(1 IN°= 80°0 AMPS)
 NPM(1 IN°=200°0 RPM)
 EB(1 IN°=200°0 VOLTS)
 AI(1 IN°= 2000°0 AMPS)

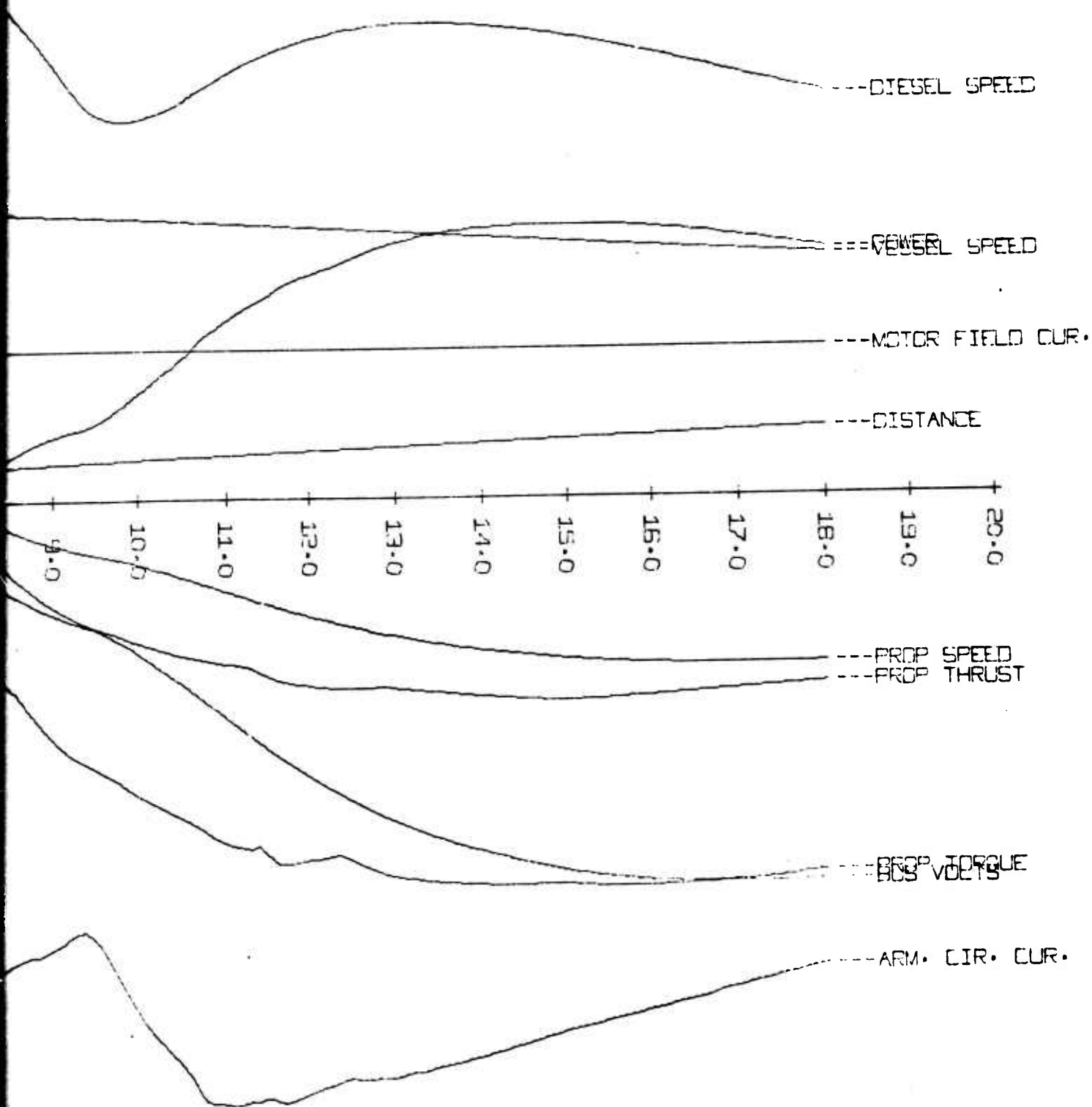
SERIAL NUMBER-- 1.

SIMULATION OF TEST SHIP FOR DOCUMENTATION

CRASH REVERSAL OF A 12000 TON SHIP







APPENDIX

SYMBOL TABLE

<u>FORTTRAN LABEL</u>	<u>MATH SYMBOL (if any)</u>	<u>**</u>	<u>DESCRIPTION</u>
RO	ρ		Mass density of salt water equal to 1.99
PI	π		Constant equal to 3.14159
IZB		✓	Number of propeller blades
DAR	DAR	✓	Developed area ratio of propeller
PDR	P/D	✓	Pitch-diameter ratio of propeller
D	D	✓	Propeller diameter
XJPA	J_{pair}	✓	Propeller inertia in air
XJM	J_m	✓	Motor inertia
XJSH	J_{sh}	✓	Shaft inertia
XJD		✓	Diesel inertia
XJG		✓	Generator inertia
EFG	e_{fg}		Ordered generator field voltage
XJMP	J_{mp}		Inertia of propeller-motor system
XJDG	J_{pmg}		Combined diesel-generator inertia
FIMAX	i_{fmax}	✓	Maximum value of motor field current
RA	R_a		Armature circuit resistance
RG	R_g	✓	Generator armature resistance
C1	C_1	✓	} Constants used in calculating ordered armature current in motor regulator
XK1	K_1	✓	
QBRG		✓	Y-intercept of developed diesel torque vs. angular velocity of prime mover curve
CSHFR	C_2	✓	Shaft friction constant
CWM		✓	Slope of developed diesel torque vs. angular velocity of prime mover curve
PCMS		✓	Entrained water inertia

**a check(✓) implies the variable is initially read in by subroutine REDAT.

<u>FORTTRAN LABEL</u>	<u>MATH SYMBOL (if any)</u>	<u>**</u>	<u>DESCRIPTION</u>
XLA	L	✓	Armature circuit inductance
TAU1	τ_1	✓	Time for bridge controller to reach position θ_1 from initial position θ_{ic}
TAU2	τ_2	✓	Time for bridge controller to go from position θ_1 to the final (ordered) position θ_0
XK6	K_6	✓	Developed diesel torque constant
ALP1	α_1	✓	Buffer spring scale
ALP ₂	α_2	✓	Governor to engine constant
ALP ₃	α_3	✓	Needle valve adjustment
ALPD	α_d	✓	Speed droop coefficient
XMASS			Used to contain the value $2240 \cdot (1 + C_{ad})$ $\Delta/32.18$ used in the computation of the ship acceleration.
NØVAR			(Graph routine) indicates number of variables to be plotted
NVAR			(Graph routine) 10-word array con- taining the code numbers of up to 10 variables to be plotted
JJ			(Graph routine) index to NVAR array
ICNTR		✓	Indicates particular command given; equal to one (1) for ahead flank, 2 for back flank, 3 for ahead full, and 4 for ahead full in ice.
NPLOT			(Graph routine) indicates if a grid is to be plotted; equal to zero (0) if a grid not desired, equal to one (1) if a grid is requested.

** a check (✓) implies the variable is initially read in by subroutine REDAT.

<u>FORTTRAN LABEL</u>	<u>MATH SYMBOL (if any)</u>	<u>**</u>	<u>DESCRIPTION</u>
NDATA			(Not used)
NON		✓	Indicates the number of time intervals between listings of the output on the printer.
NIP			(Graph routine) index to NVAR array
SX		✓	(Graph routine) plotter scale factor for time axis
SY			(Graph routine) plotter scale factor for variable currently being plotted.
SY1		✓	(Graph routine) plotter scale factor for armature circuit current
SY2		✓	(Graph routine) plotter scale factor for bus volts
SY3		✓	(Graph routine) plotter scale factor for diesel speed
SY4		✓	(Graph routine) plotter scale factor for motor field current
SY5		✓	(Graph routine) plotter scale factor for propeller speed
SY6		✓	(Graph routine) plotter scale factor for power
SY7		✓	(Graph routine) plotter scale factor for vessel speed
SY8		✓	(Graph routine) plotter scale factor for propeller torque
SY9		✓	(Graph routine) plotter scale factor for propeller thrust.

** a check (✓) implies the variable is initially read in by subroutine REDAT.

<u>FORTTRAN</u> <u>LABEL</u>	<u>MATH</u> <u>SYMBOL</u> <u>(if any)</u>	<u>**</u>	<u>DESCRIPTION</u>
SY10		✓	(Graph routine) plotter scale factor for ship's distance
SY11		✓	(Graph routine) plotter scale factor for ice torque
SY12		✓	(Graph routine - not used)
DT	dt	✓	Time interval
CT			Save area for current time
T	t	✓	Current time
TEND		✓	Time ending value
TIMSC		✓	Time increment for writing values on the disk
Z	Z	✓	Servo stroke position
Y	dz/dt	✓	Servo stroke velocity
XNPM	n_{pm}	✓	Angular velocity of prime mover
FGI	i_{fg}	✓	Generator field current
FIP	i'_f	✓	Motor field adjusting current
XNP	n_p	✓	Propeller angular velocity
VS	V_s	✓	Ship speed
XS		✓	Ship distance
SIE			Angular position of propeller
AI	i_a	✓	Armature circuit current
RV		✓	(Not used)
TEMP		✓	(Not used)
CZ			These 12 variables are related to the 12 variables immediately above (relationship indicated by matching labels after dropping the initial C)
CY			
CXNPM			
CFGI			

** a check (✓) implies the variable is initially read in by subroutine REDAT.

<u>FORTTRAN LABEL</u>	<u>MATH SYMBOL (if any)</u>	<u>**</u>	<u>DESCRIPTION</u>
CFIP			(continued from previous page) and are used to save the values of the above variables at each step when the next set of values are being calculated. (CRV and CTEMP are not used)
CXNP			
CVS			
CXS			
CSIE			
CAI			
CRV			
CTEMP			
ZDO	dz/dt		Servo stroke velocity
YDO	d^2z/dt^2		Servo stroke acceleration
XNPMD	dn_{pm}/dt	✓	Time rate of prime mover speed
FGIDO	di_{fg}/dt		Time rate of generator field current
FIPDO	di'_f/dt		Time rate of motor field adjusting current
XNPDO	dn_p/dt		Propeller acceleration
VSDO	dV_s/dt		Ship's acceleration
XSDO	V_s		Ship's velocity
SIEDO	n_p		Propeller angular velocity
AIDO	di_a/dt	✓	Time rate of armature circuit current
RVDO			(Not used)
TEMDO			(Not used)
XNE	n_e		Diesel error speed
XNEDO	dn_e/dt		time rate of diesel error speed
VA	V_a		Advance velocity of propeller
THETO	θ^*		Ordered (final) bridge lever position
THET1	θ_1	✓	Bridge lever position reached in τ_1 seconds

** a check (✓) implies the variable was initially read in by subroutine REDAT.

<u>FORTTRAN LABEL</u>	<u>MATH SYMBOL (if any)</u>	<u>**</u>	<u>DESCRIPTION</u>
THETA	θ		Actual bridge level position
THETD	$d\theta/dt$		Time rate of bridge lever position
THEIC	θ_{ic}	✓	Initial value of bridge lever position
QDE	Q_{de}		Developed diesel torque
QG	Q_g		Generator torque
EG	e_g		Generator voltage
EB	e_b		Bus voltage
EC	e_c		Back emf
QM	Q_m		Motor torque
QFRM	Q_{frm}		Shaft friction torque
AIHAT	\hat{i}_a		Ordered armature circuit current
QP	Q_p		Propeller torque
QIP			Ideal propeller torque
QI			Ice torque
TP	T_p		Total propeller thrust
XPROI	T		Thrust per propeller
R	R		Ship's resistance
VAR			(Graph routine) contains variable retrieved from the disk
XKW			Ship's power in kilowatts
CVK			Ship's speed in knots
CFG	C_{fg}		Generator flux
SPGEN		✓	Spacing of generator, independent axis for CFGA array
CFGA		✓	13-word array containing generator flux for spacing indicated by SPGEN
TRNFG	N_{fg}	✓	Number of turns in generator
ZGP	Z'_g	✓	Generator flux conversion factor

** a check (✓) implies the variable was initially read in by subroutine REDAT.

<u>FORTTRAN LABEL</u>	<u>MATH SYMBOL (if any)</u>	<u>**</u>	<u>DEFINITION</u>
RFG	R_{fg}	✓	Generator field resistance
XLFG	L_{fg}		Generator field inductance
TAUFG	τ_{fg}		Generator time constant
TRNSF	N_{sf}	✓	Number of turns in generator field
RX		✓	Variable resistance in generator field
XM			Mutual inductance between generator separate and in series fields
XLASF	L_{asf}		Inductance of generator series field
AMPIR			Ampere turns of generator
XGEN		✓	Number of generators
CFM	C_{fm}		Motor field flux
SPMOT		✓	Spacing of motor, independent axis for CFMA array
CFMA		✓	13-word array containing motor flux for spacing indicated by SPMOT
TRNFM	N_{fm}	✓	Number of turns in motor
ZMP	Z'_m	✓	Motor flux conversion constant
RFM	R_{fm}	✓	Motor field resistance
XLFM	L_{fm}		Motor field inductance
TAUFM	τ_{fm}		Motor field time constant
FMIRV		✓	Field current increment for reverse direction
TAUA	τ_A	✓	Time constants controlling how fast FMIRV gets added to FMI
TAUB	τ_B	✓	
B	K_T	✓	64-word array containing propeller thrust coefficients using J
BE	K_Q	✓	64-word array containing propeller torque coefficients using J

** a check (✓) implies the variable was initially read in by subroutine REDAT.

<u>FORTTRAN</u> <u>LABEL</u>	<u>MATH</u> <u>SYMBOL</u> <u>(if any)</u>	<u>**</u>	<u>DESCRIPTION</u>
E	C_T	✓	32-word array containing propeller thrust coefficients using $1/J$
EE	C_Q	✓	32-word array containing propeller torque coefficients using $1/J$
EHP		✓	20-word array containing effective horsepower for hull
WDED	(1-w)		Wake deduction for hull
WADED		✓	20-word array of wake deduction factors for hull
TDED	(1-t)		Thrust deduction for hull
THD ^W		✓	20-word array of thrust deduction factors for hull
XKK	K_n	✓	Number of propeller shafts
IELO			Disk file record index
JELO			Controls initial computation of ice torque
KELO			Indicates position of ship in ice flow
LELO			(Not used)
MELO			(Not used)
SPARE			5-word array, the first 3 words of which are used as follows: SPARE(1) equivalenced to FMI (i_{fm}) -motor field current SPARE(2) equivalenced to XNGR(n_{gr}) -ordered diesel speed SPARE(3) equivalenced to XNGRD (dn_{gr}/dt) -ordered diesel speed rate

** a check (✓) implies the variable was initially read in by subroutine REDAT.

<u>FORTTRAN</u> <u>LABEL</u>	<u>MATH</u> <u>SYMBOL</u> <u>(if any)</u>	<u>**</u>	<u>DESCRIPTION</u>
ISPAR			5-word array, the first 2 words of which are used as follows: ISPAR(1)-controls calling of GENIS mainline from EXEC mainline ISPAR(2) -controls motor field plugging in CALVR
CAD	C_{ad}	✓	"Added mass" coefficient of hull
DELTA	Δ	✓	Vessel displacement
RM		✓	Motor armature resistance
RBA		✓	25-word array containing random depths of cut propeller blade makes in the ice as a percentage of the diameter
RBMX	R_{bx}		Maximum depth of cut the propeller blade makes in the ice
BLOC1		✓	Length of first block of ice
DBB12		✓	Distance between ice blocks 1 and 2
BLOC2		✓	Length of ice block 2
DBB21		✓	Distance between ice blocks 2 and 1 in repeating sequence
EO	e_o	✓	Tip thickness of propeller blade
EK	e_k	✓	Root thickness of propeller blade
DCT	D_{ct}	✓	Diameter of propeller hub
DCR	D_{cr}	✓	Crushing pressure of ice
DSH	D_{sh}	✓	Shear pressure of ice
CICE		✓	Constant used to increase hull resist- ance for ice conditions.

** a check (✓) implies the variable was initially read in by subroutine REDAT.

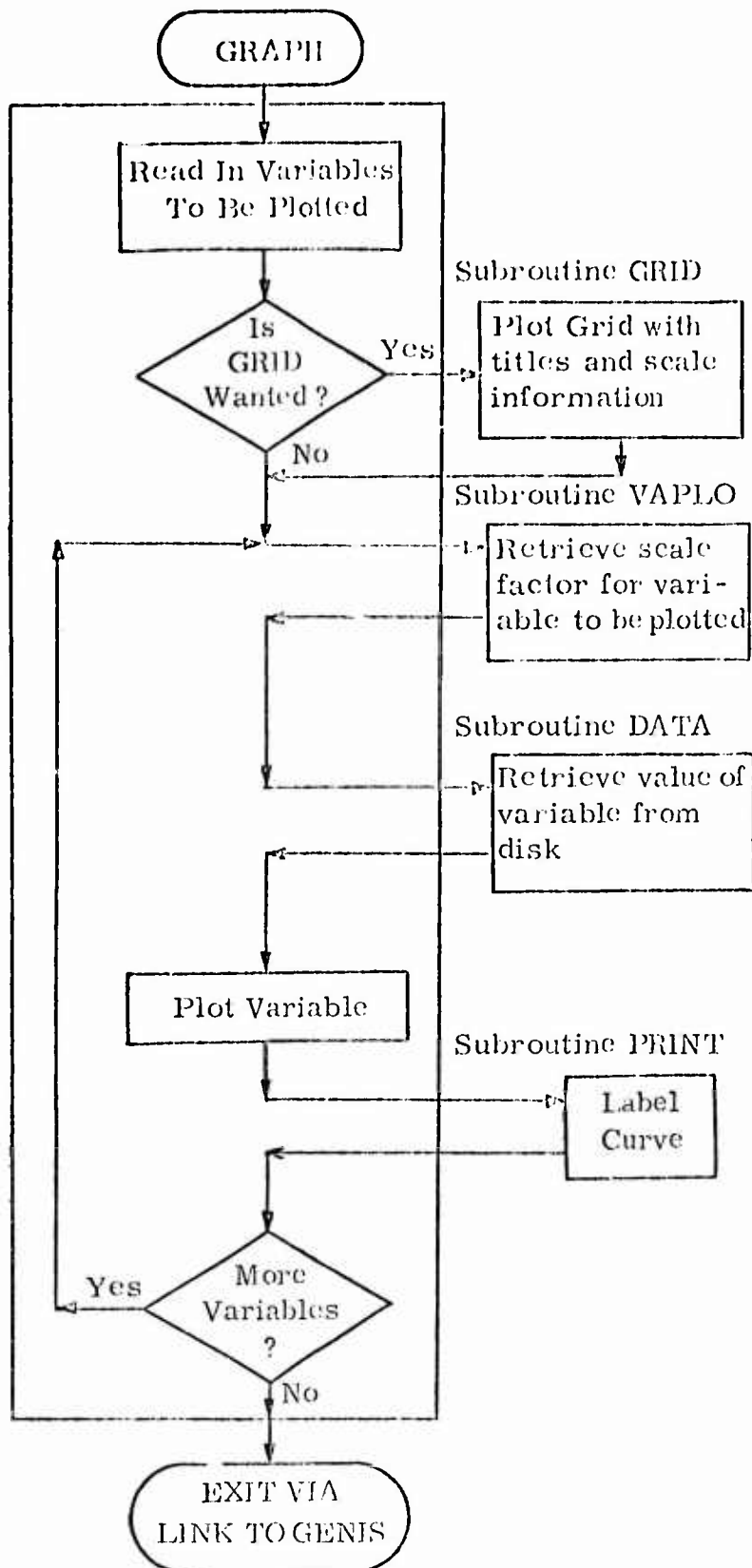
<u>FORTTRAN</u> <u>LABEL</u>	MATH SYMBOL. (if any)	<u>**</u>	<u>DESCRIPTION</u>
TICE		✓	Relative time from beginning of run until ice is encountered
CK1		✓	(Not used)
CK2		✓	(Not used)
CK3		✓	(Not used)

** a check (✓) implies the variable initially read in by subroutine REDAT.

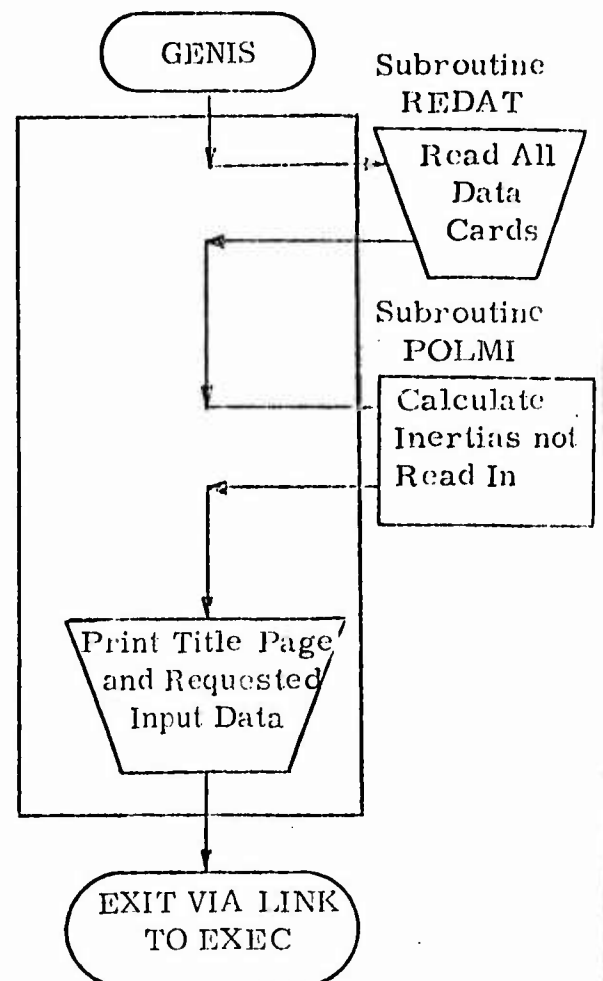
SYSTEM FLOW CHARTS

The following flow charts outline the flow of control in the three mainline programs (GENIS, EXEC, and GRAPII) during a simulation run.

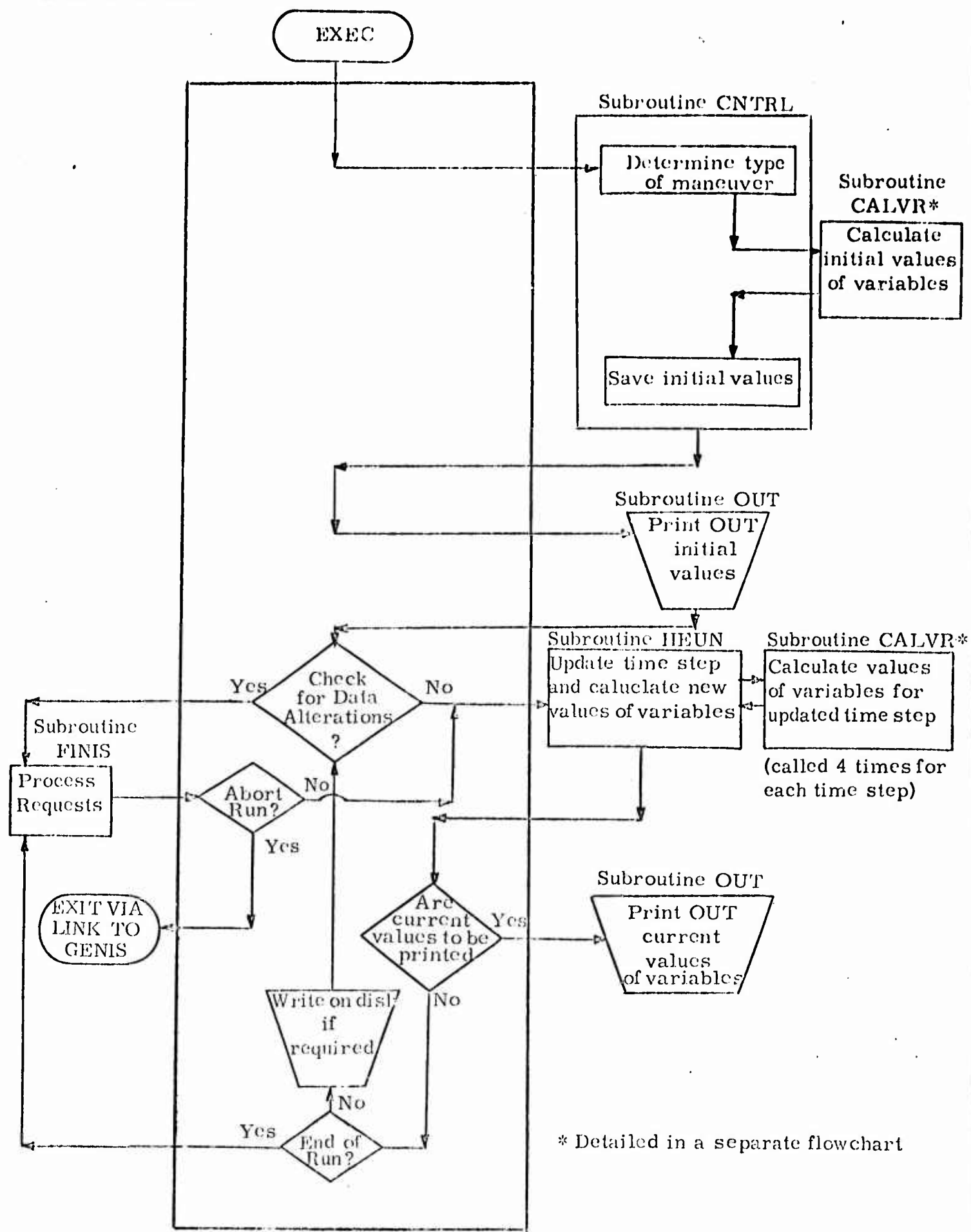
GRAPH MAINLINE



GENIS MAINLINE

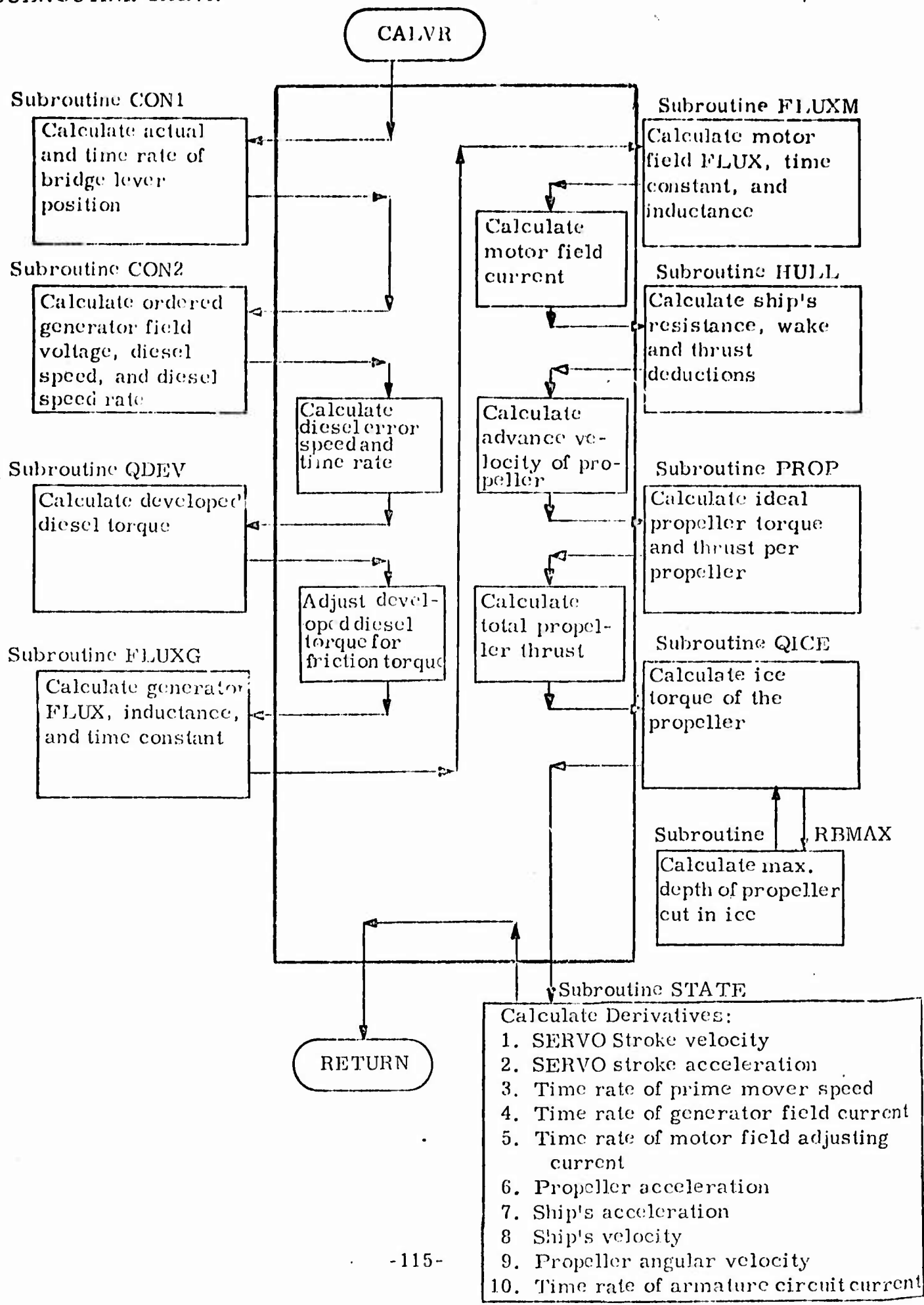


EXEC MAINLINE



* Detailed in a separate flowchart

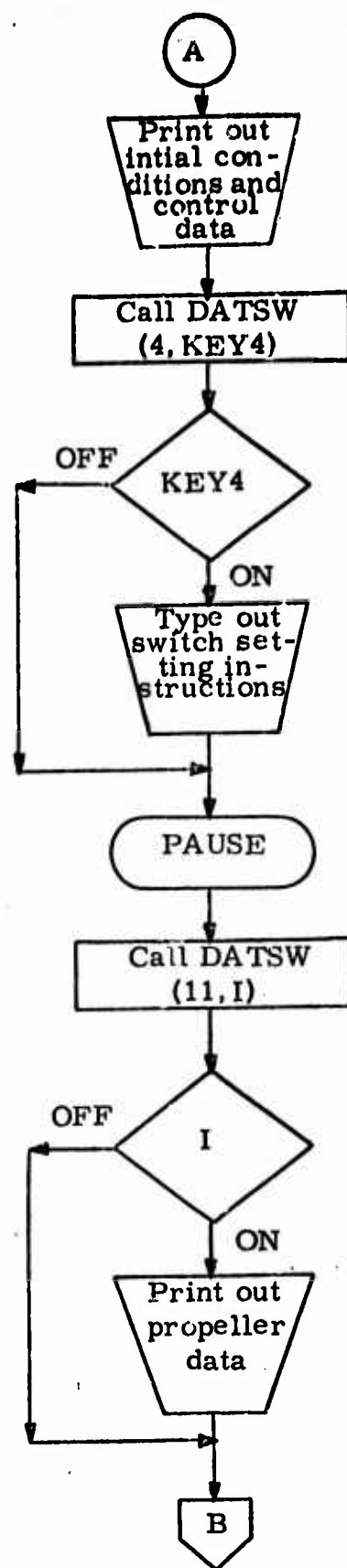
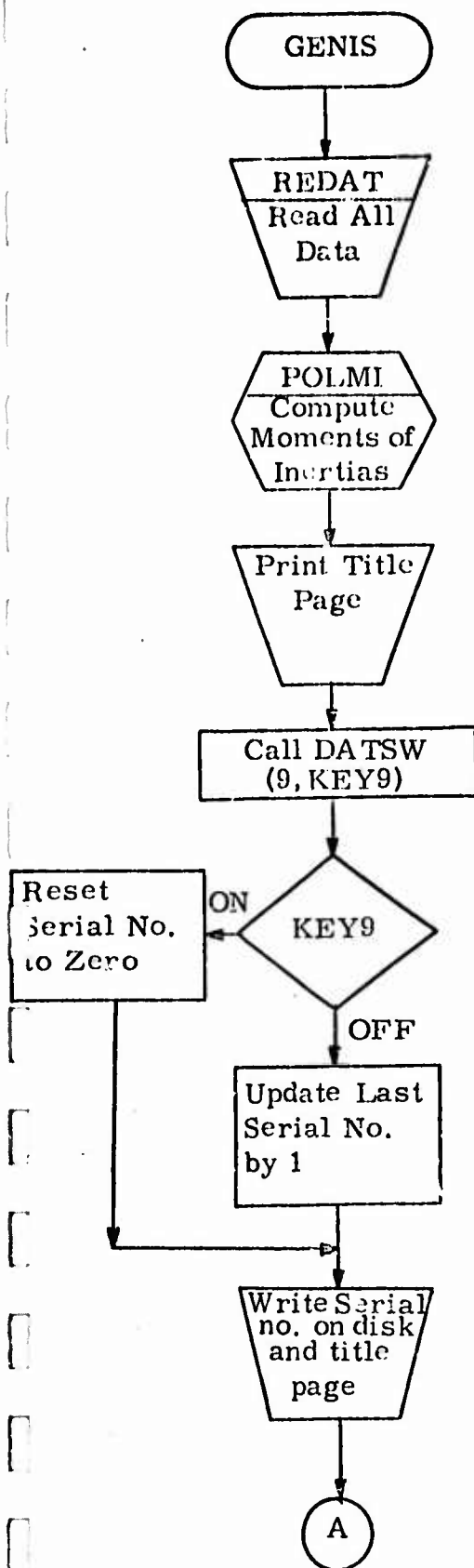
SUBROUTINE CALVR



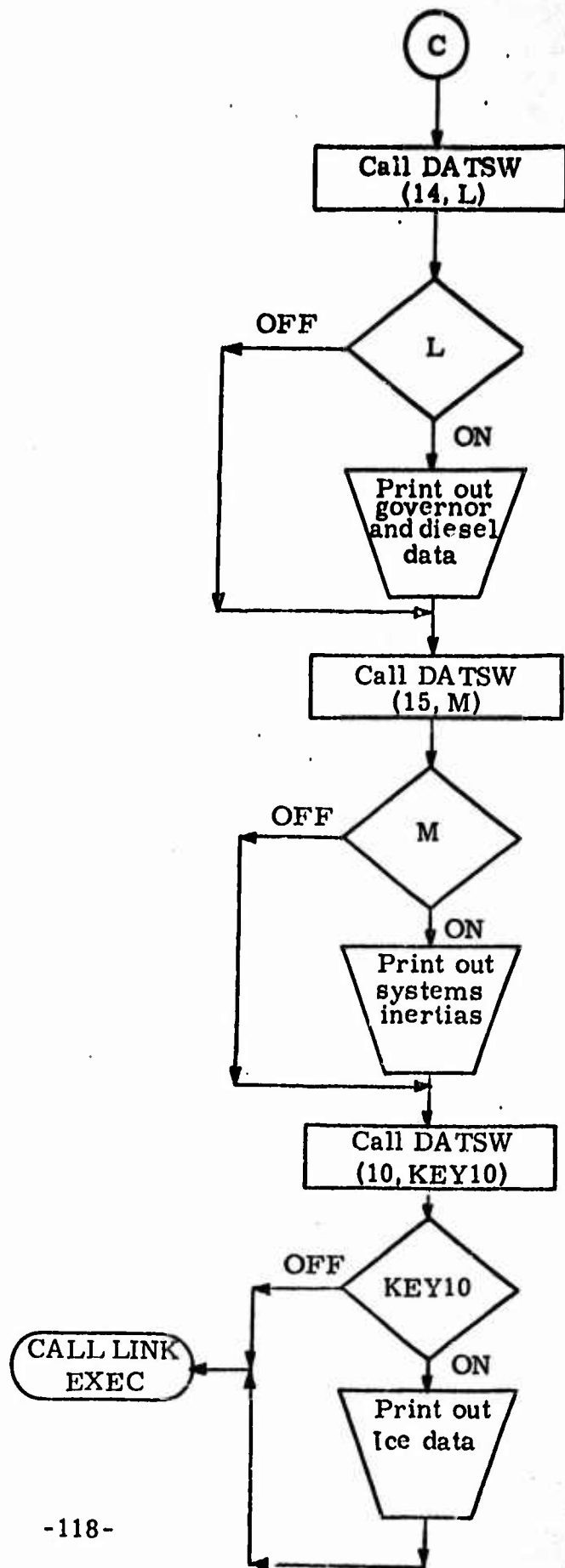
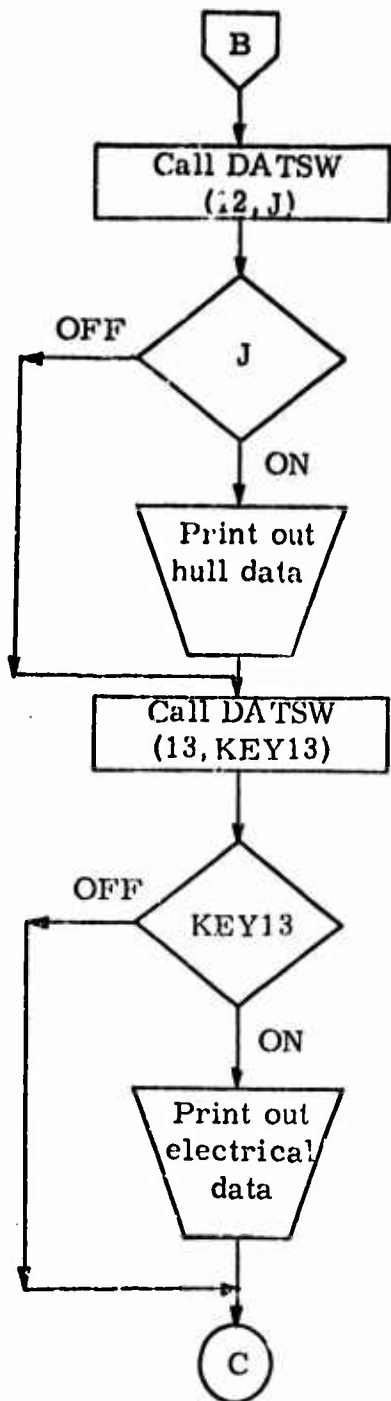
MAINLINE PROGRAMS AND SUBROUTINE
FLOW CHARTS

The following flowcharts outline in detail the processing associated with each mainline program and each subroutine.

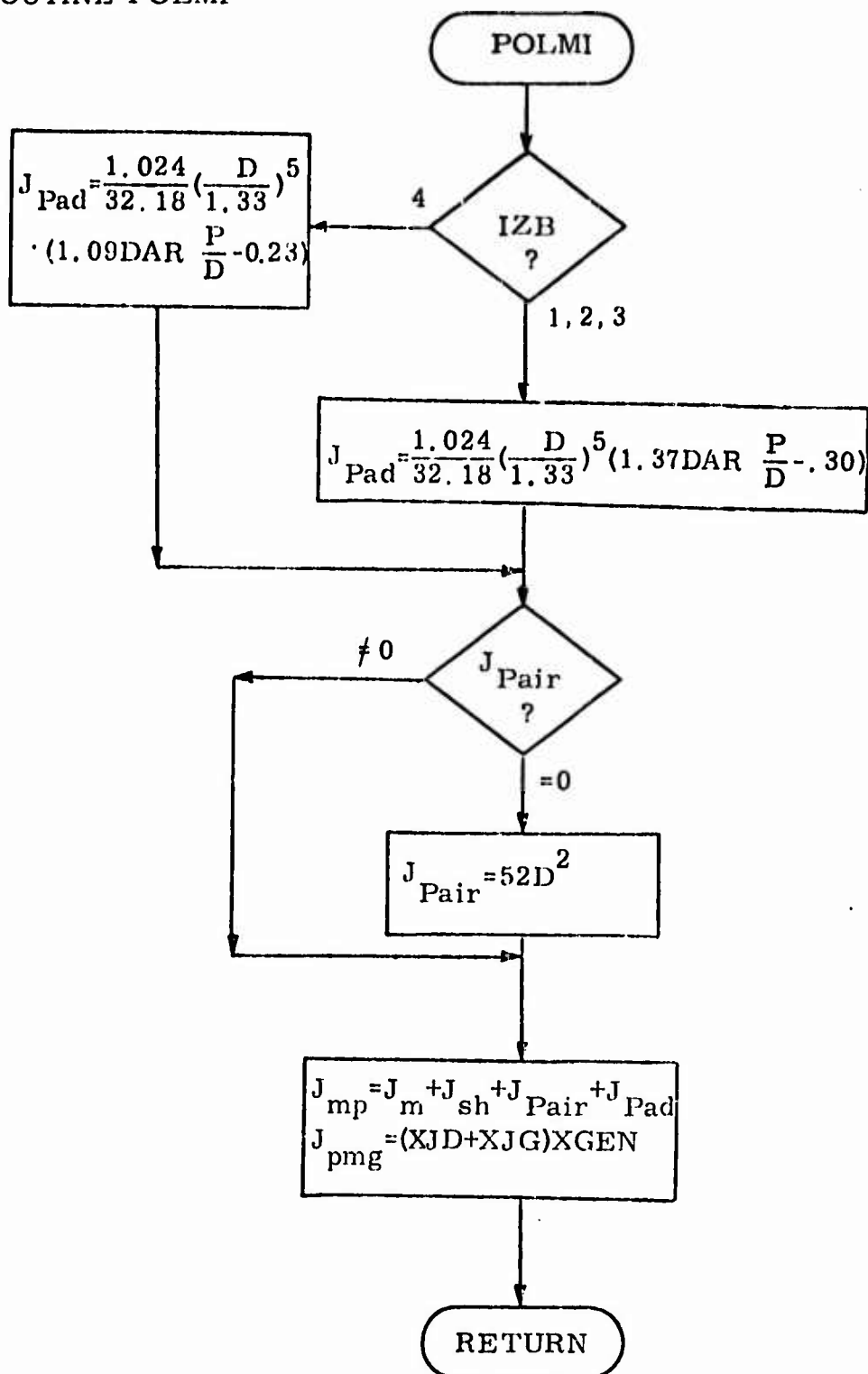
GENIS MAINLINE



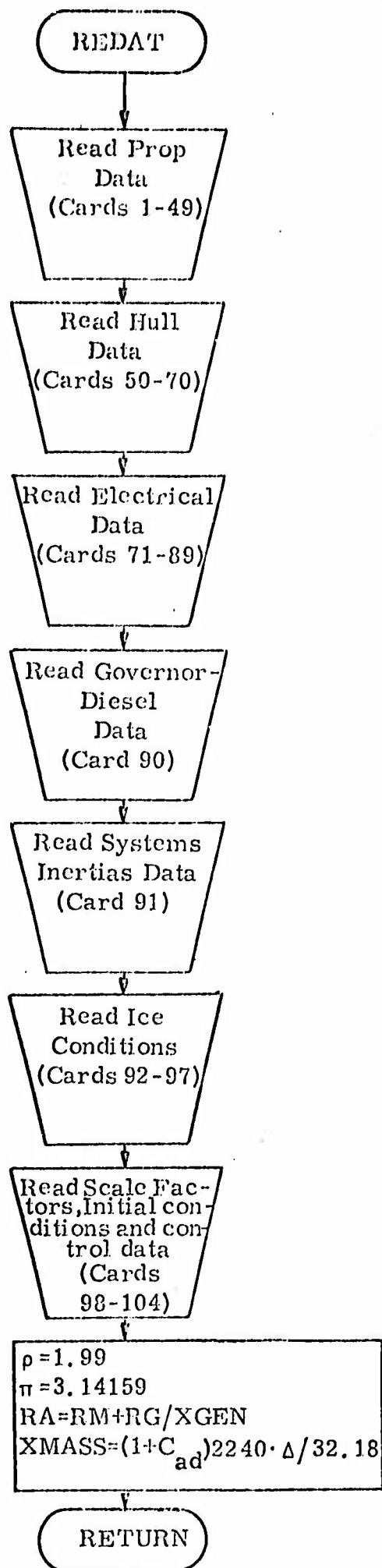
GENIS MAINLINE (continued)



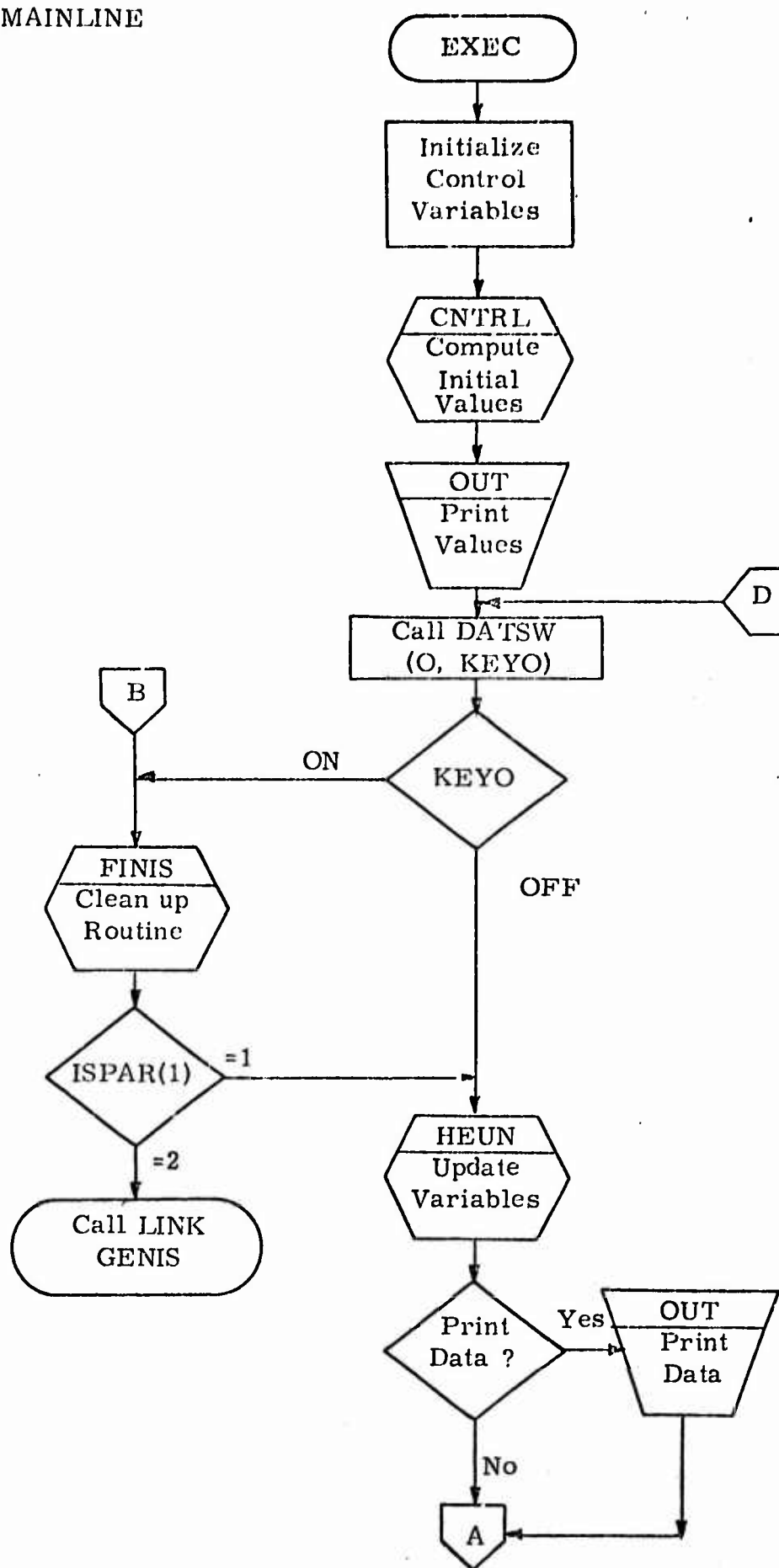
SUBROUTINE POLMI



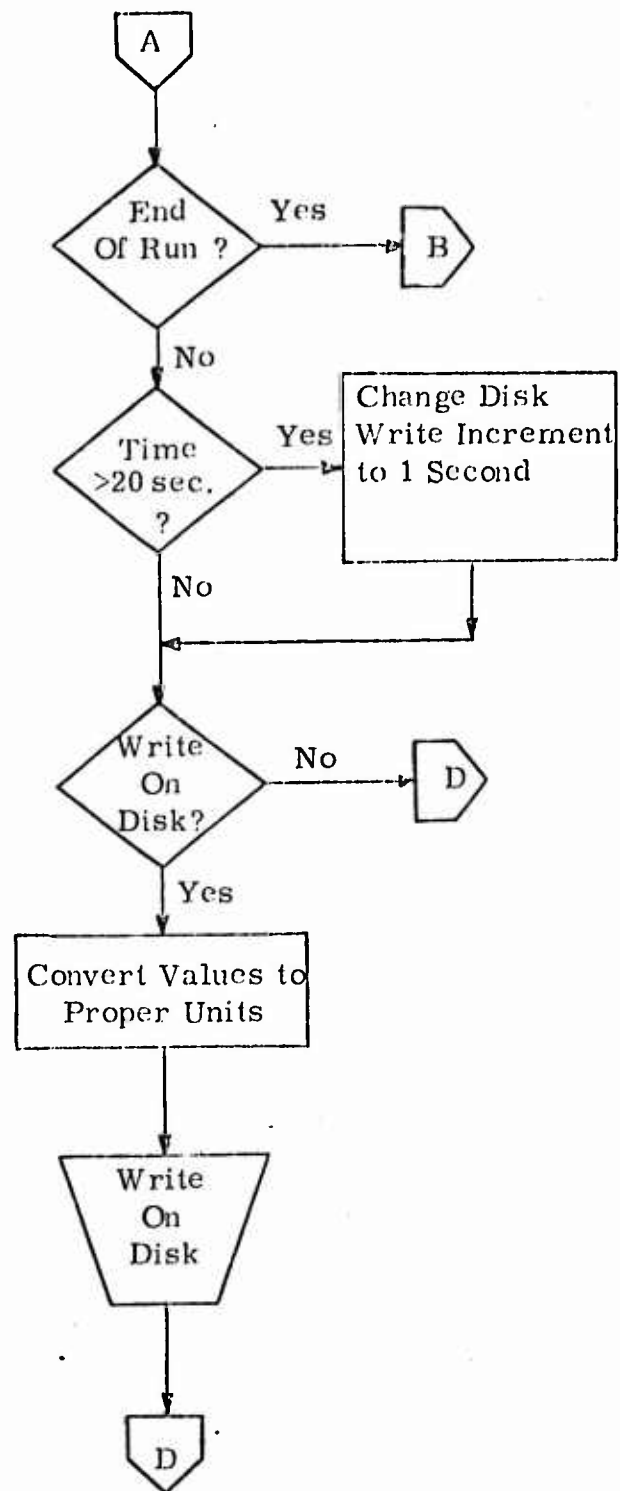
SUBROUTINE REDAT



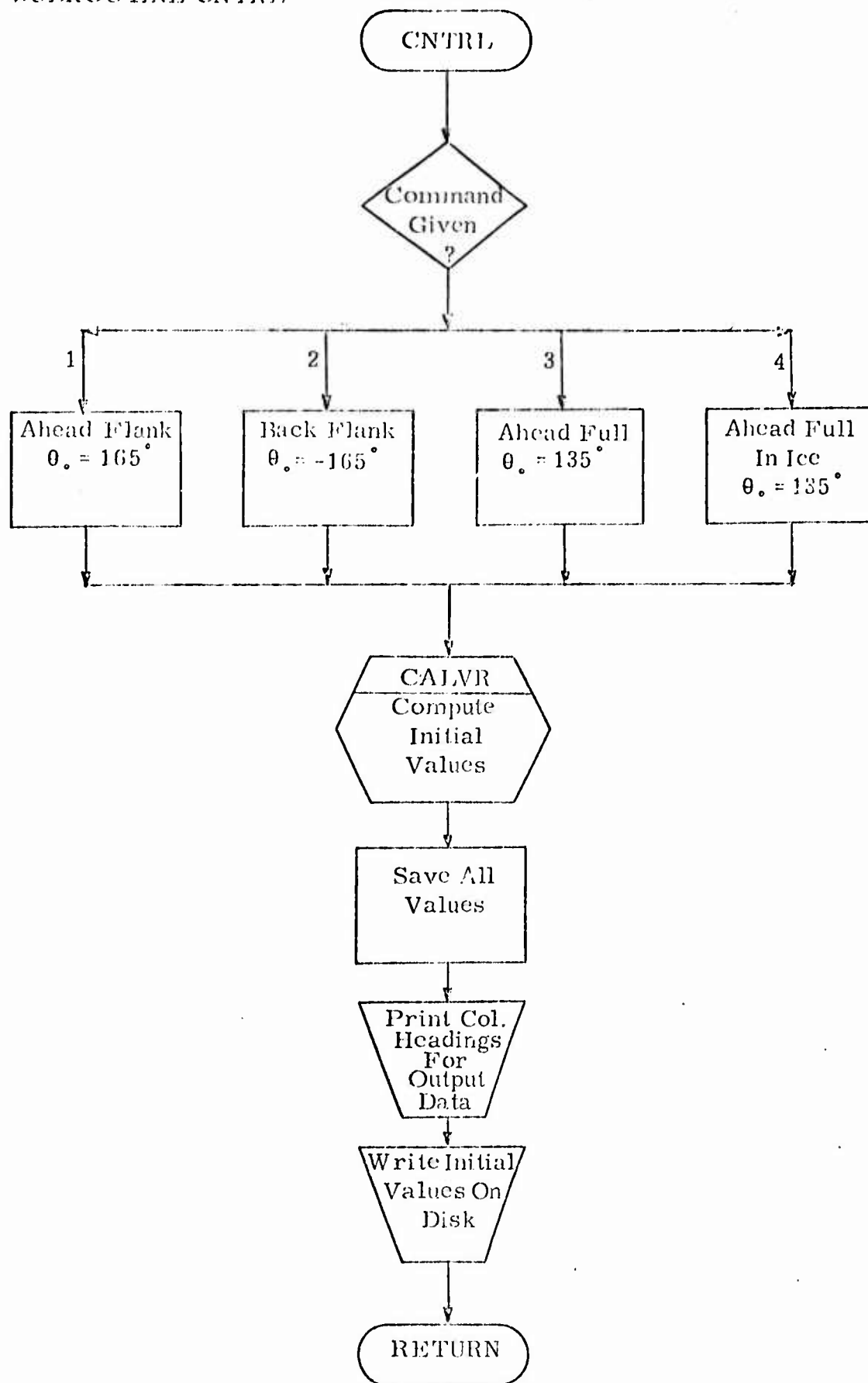
EXEC MAINLINE



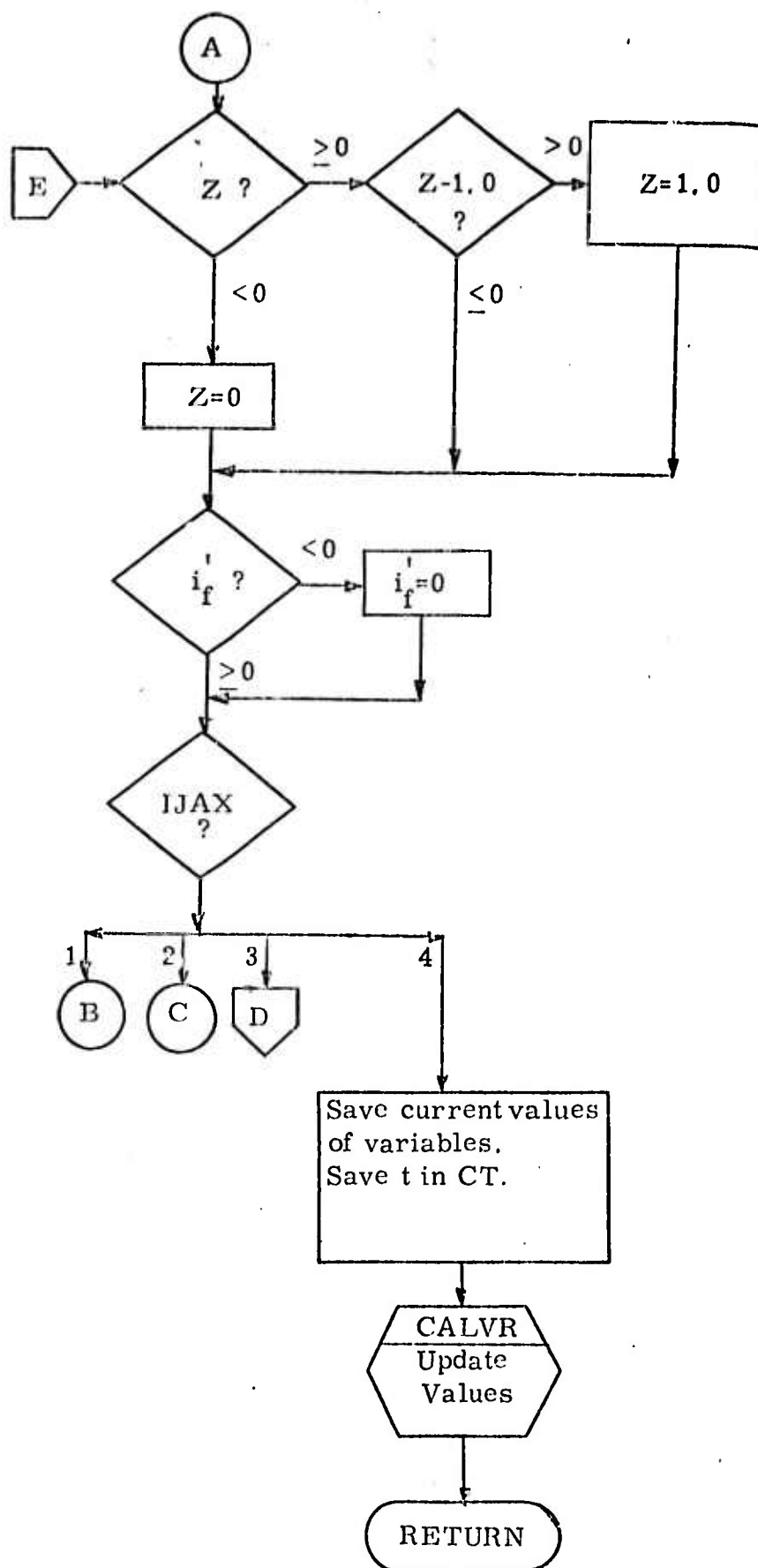
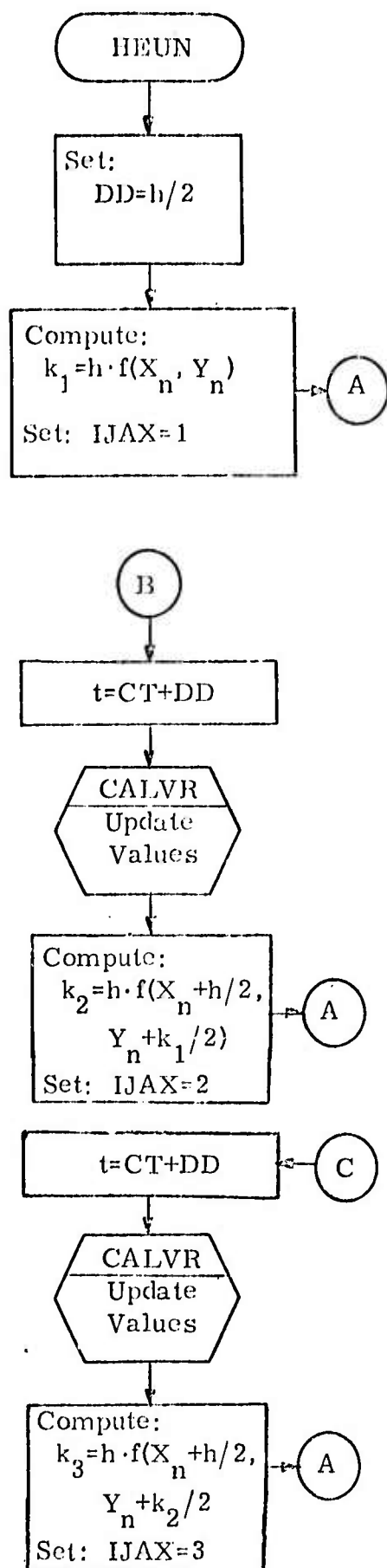
EXEC MAINLINE (continued)



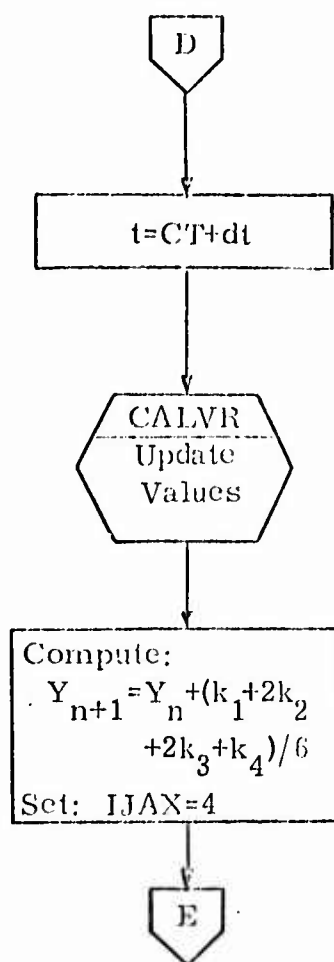
SUBROUTINE CNTRL



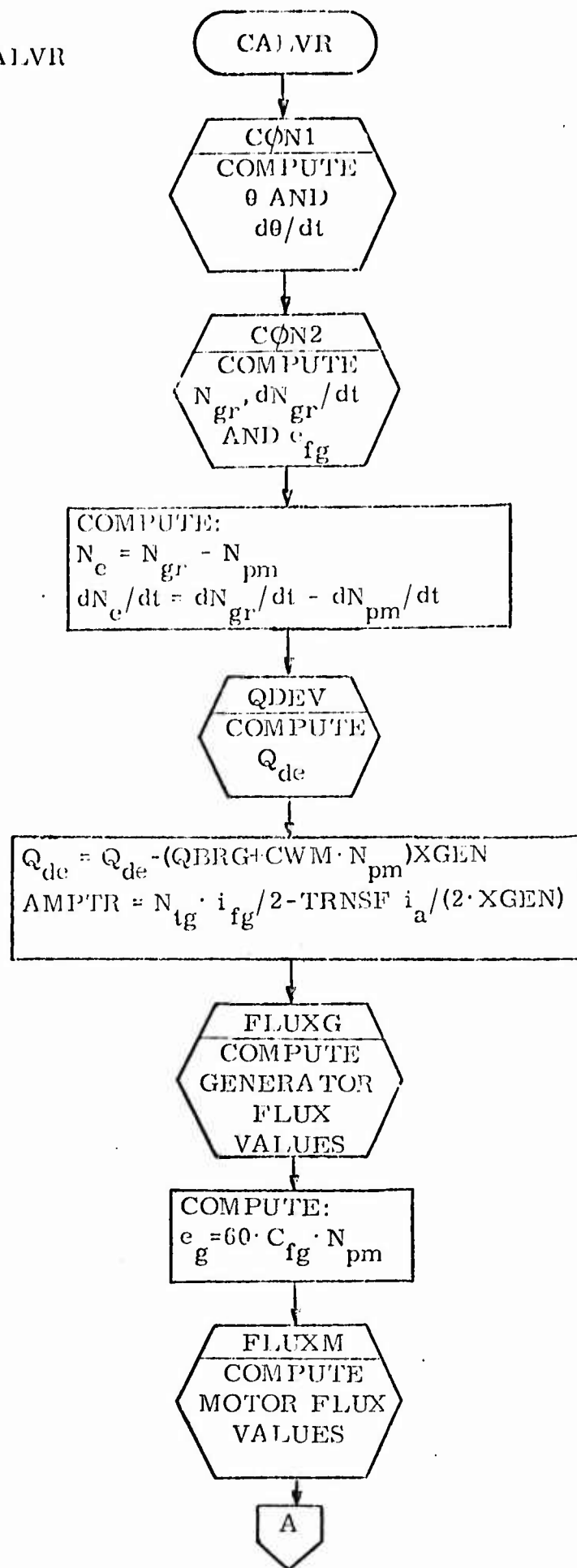
SUBROUTINE HEUN



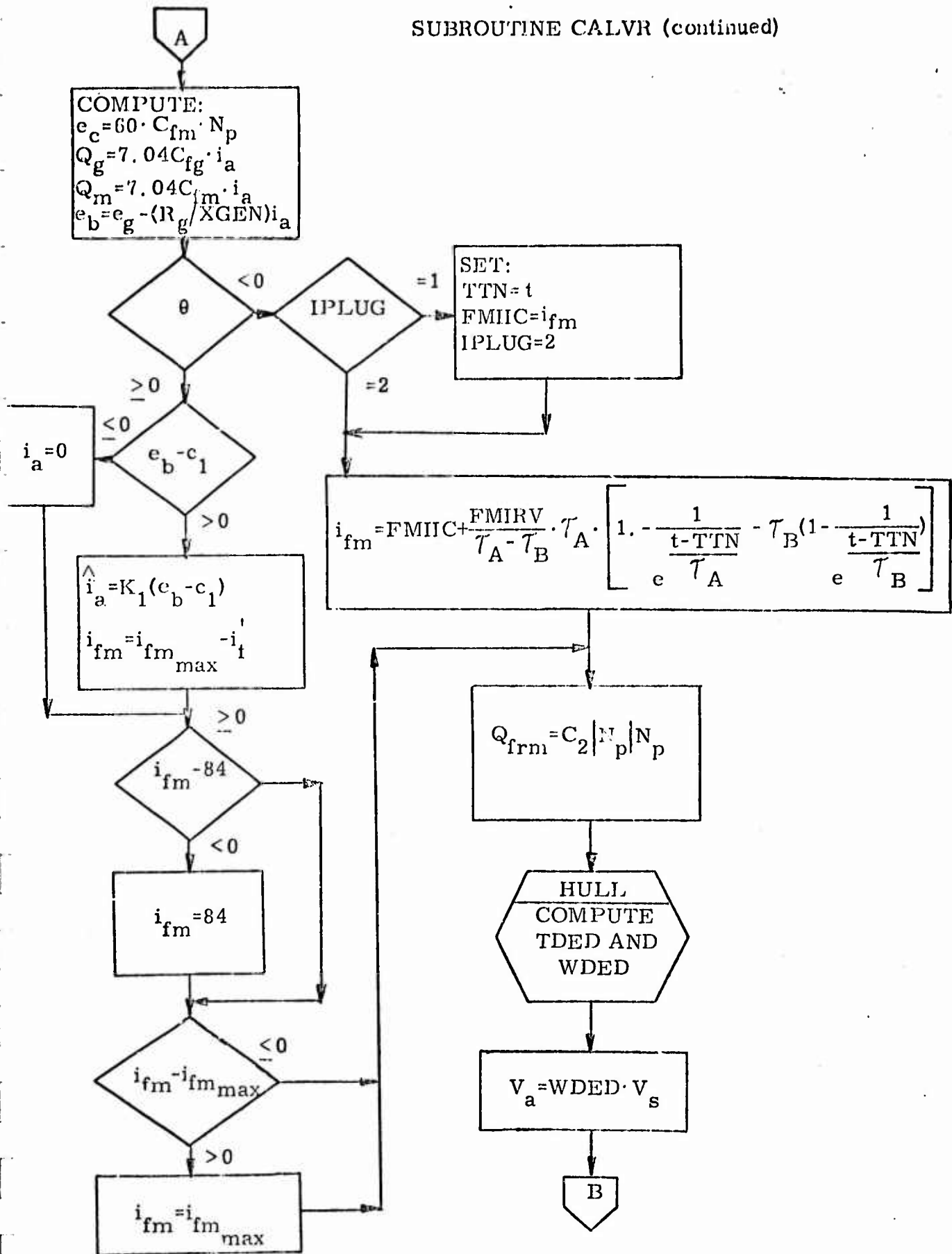
SUBROUTINE HEUN (continued)



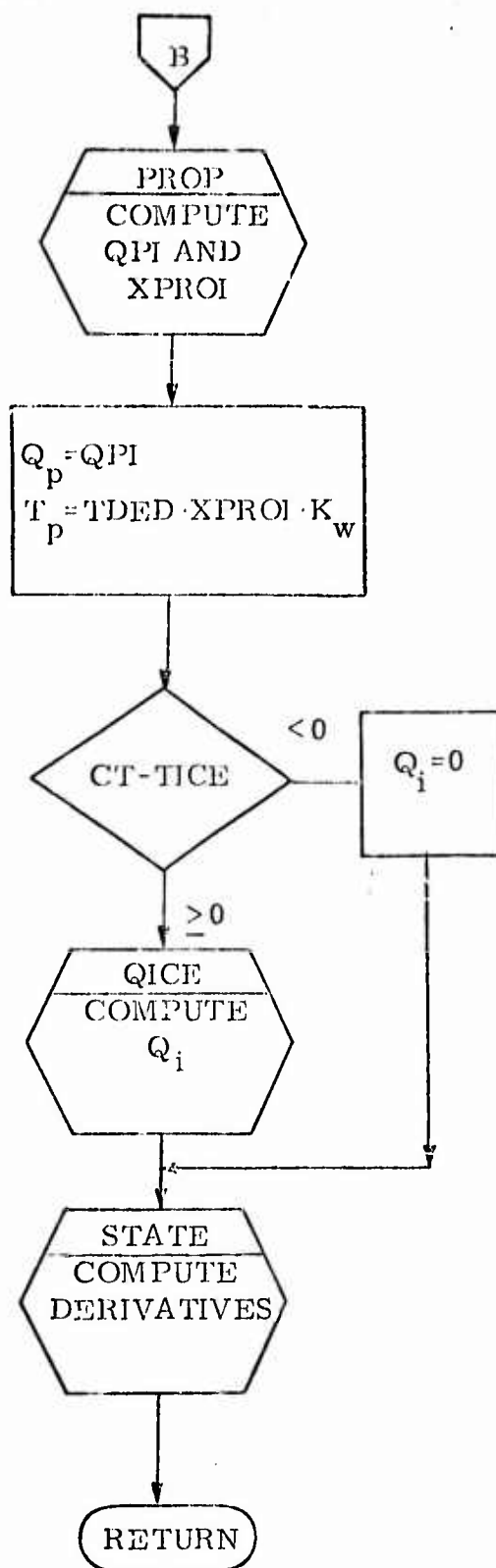
SUBROUTINE CALVR



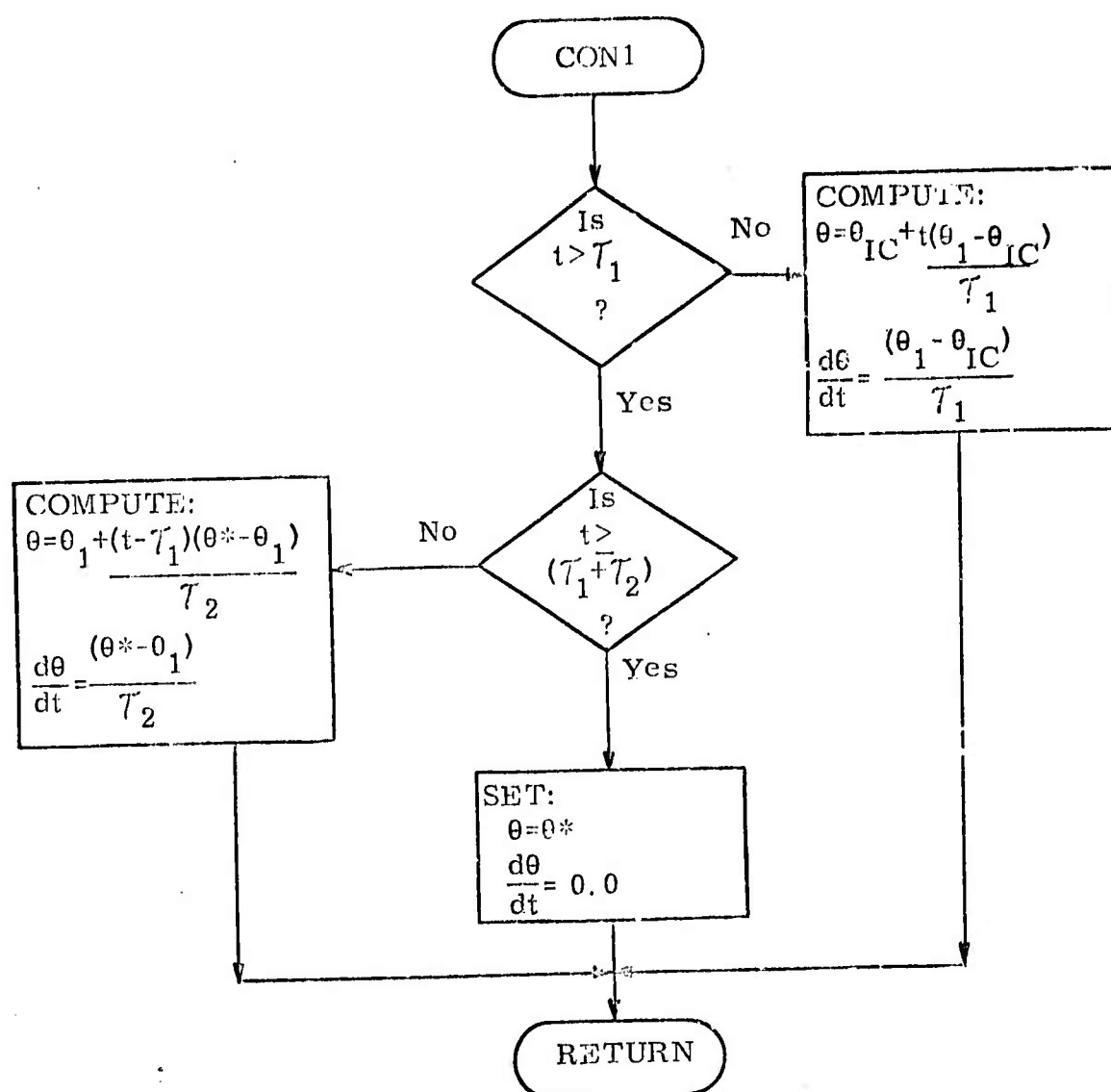
SUBROUTINE CALVR (continued)

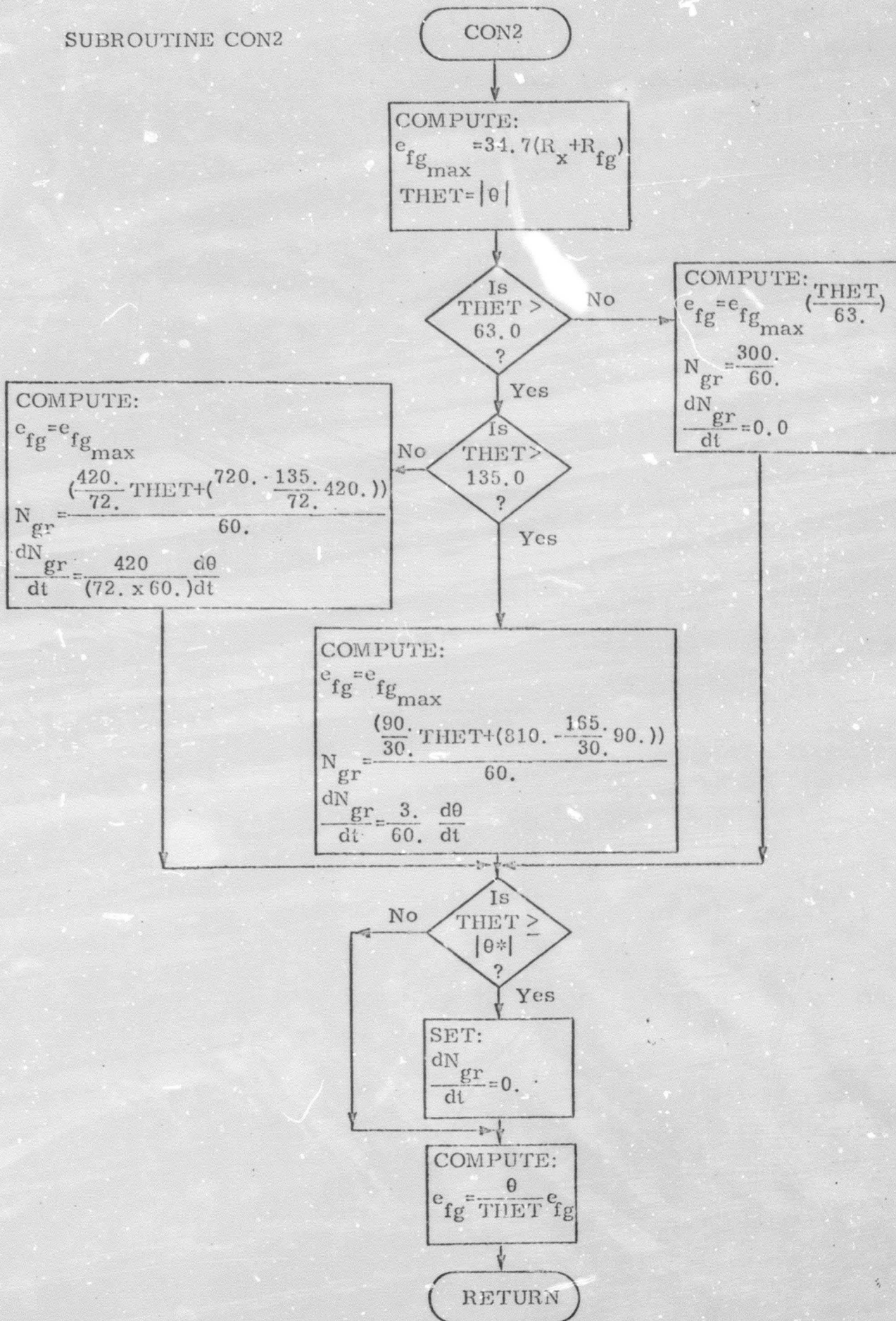


SUBROUTINE CALVR (continued)

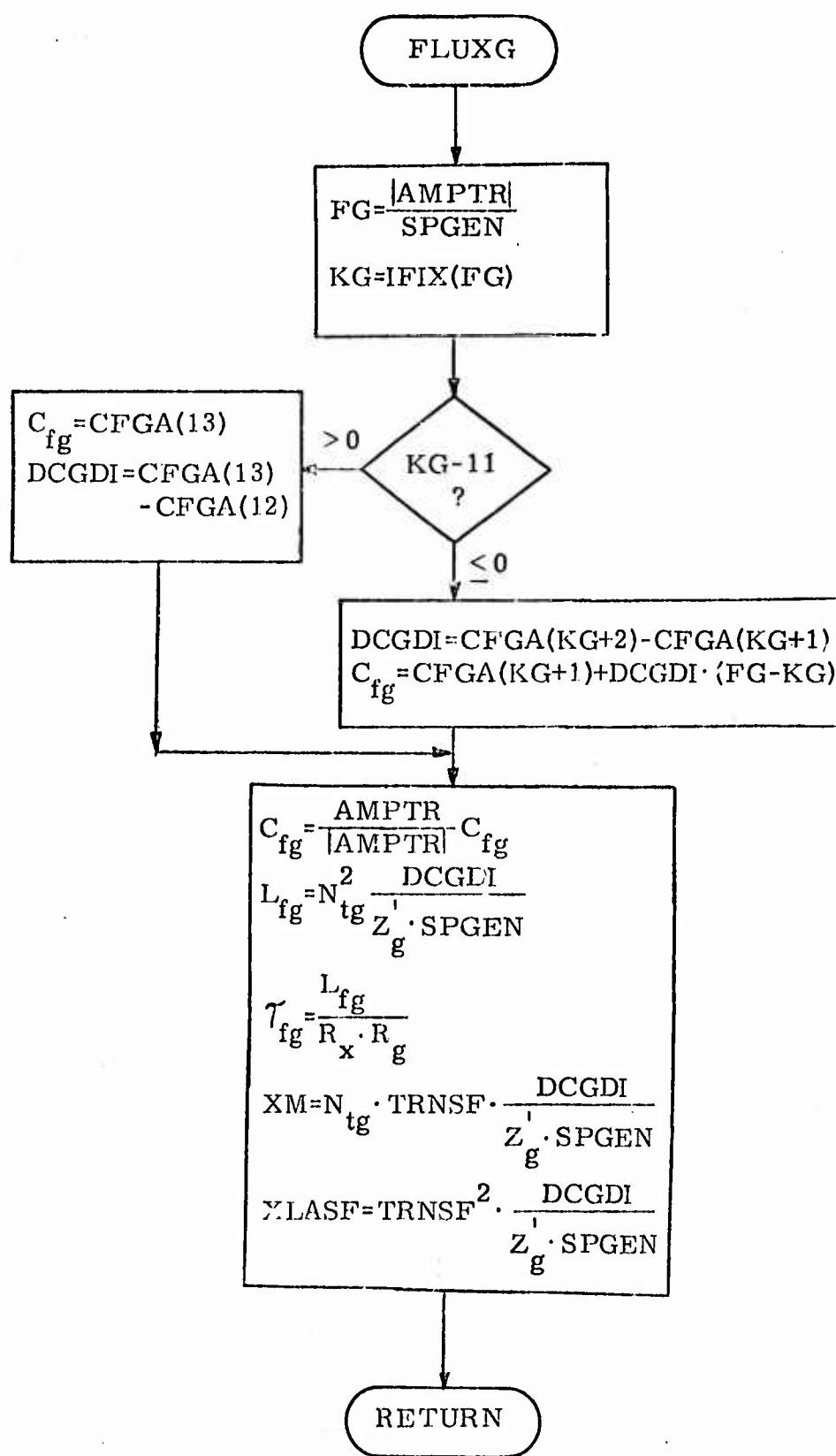


SUBROUTINE CON1

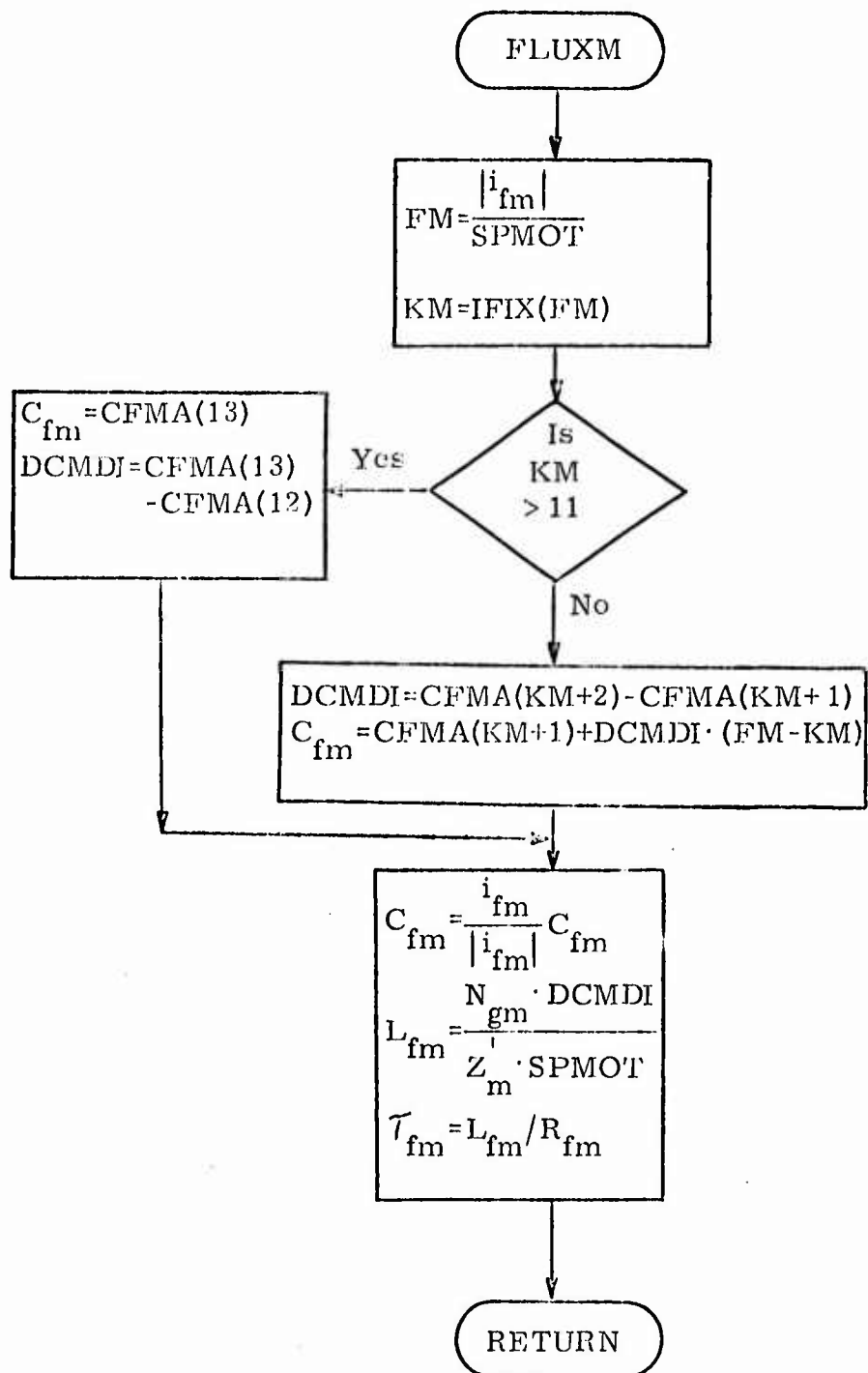




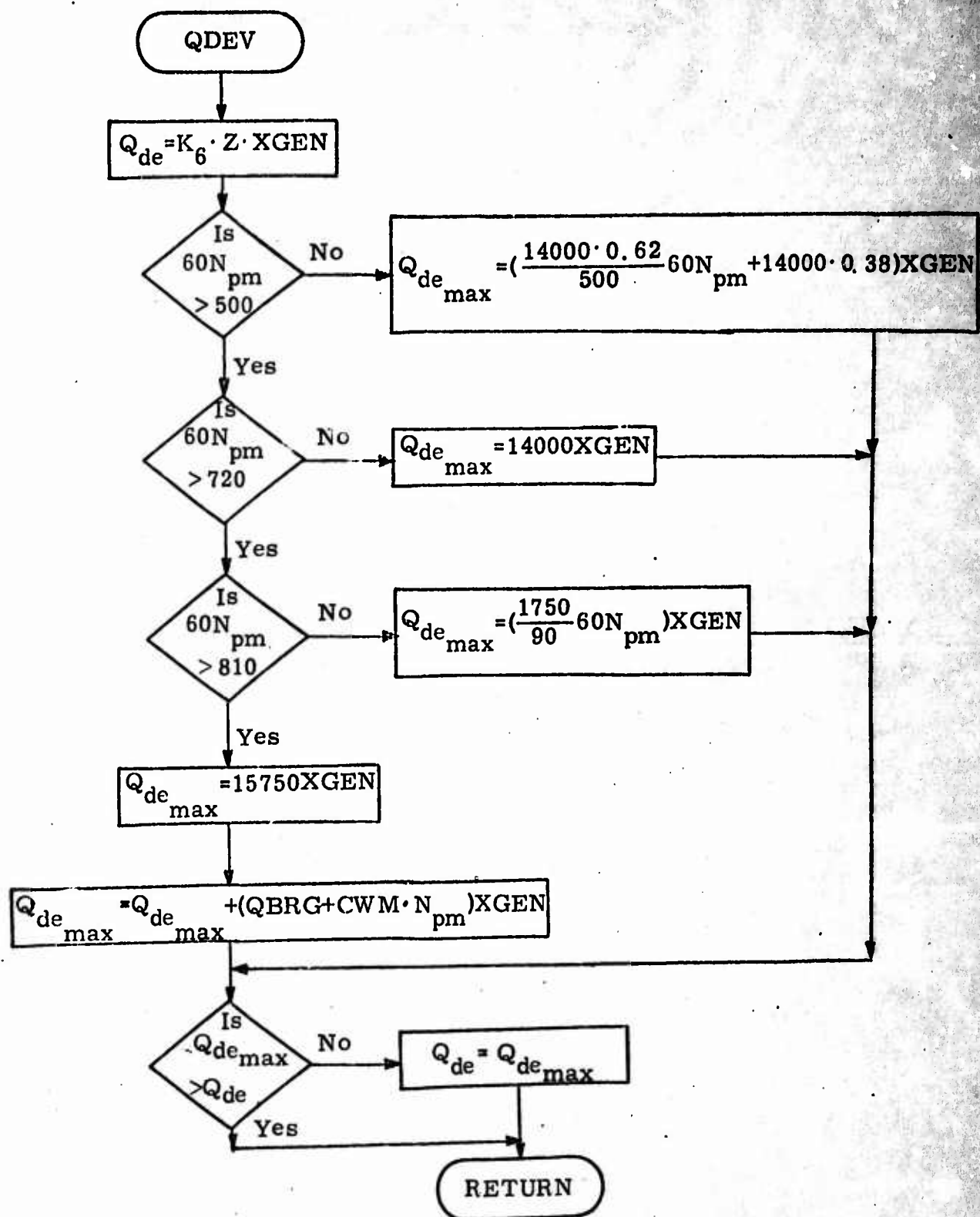
SUBROUTINE FLUXG



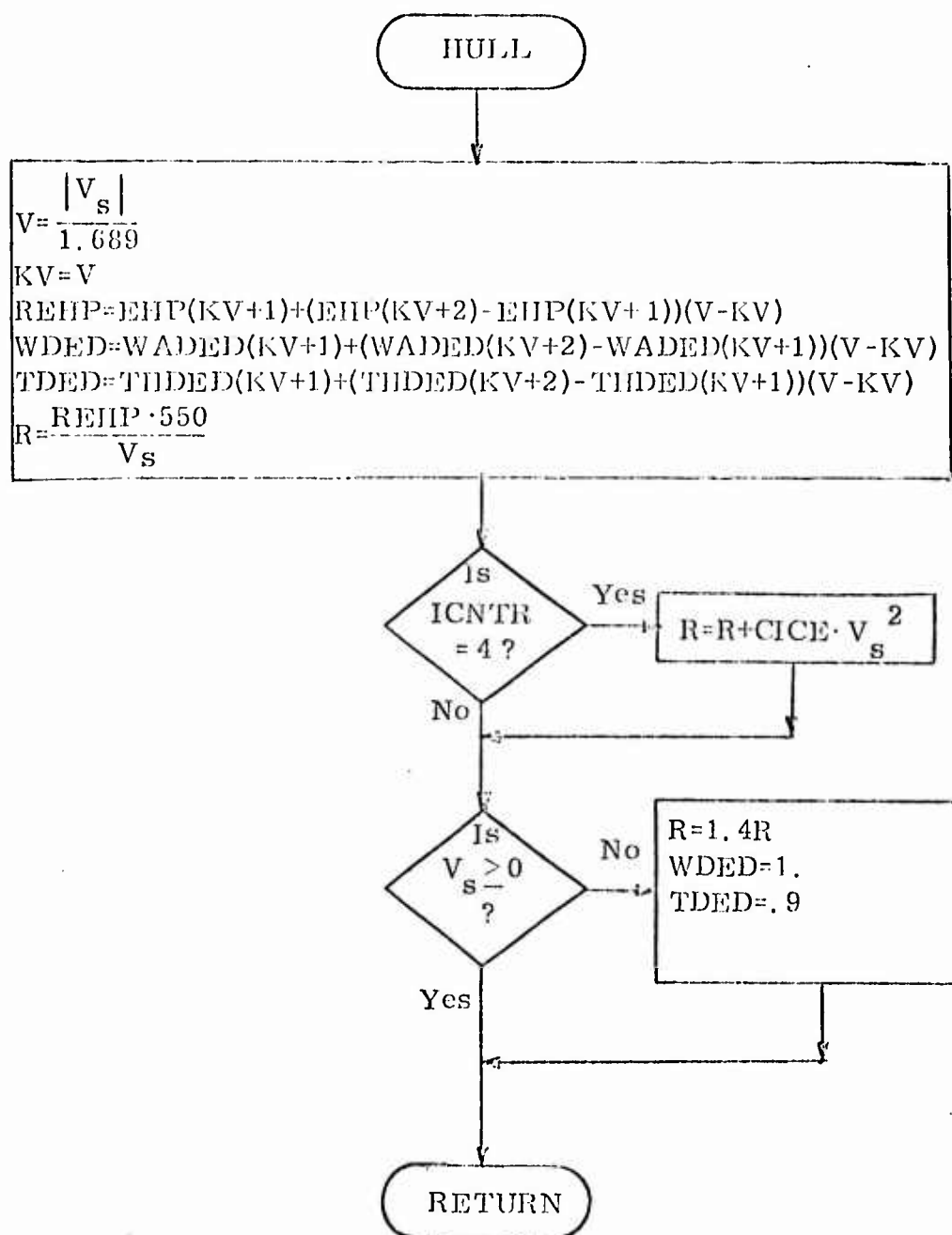
SUBROUTINE FLUXM



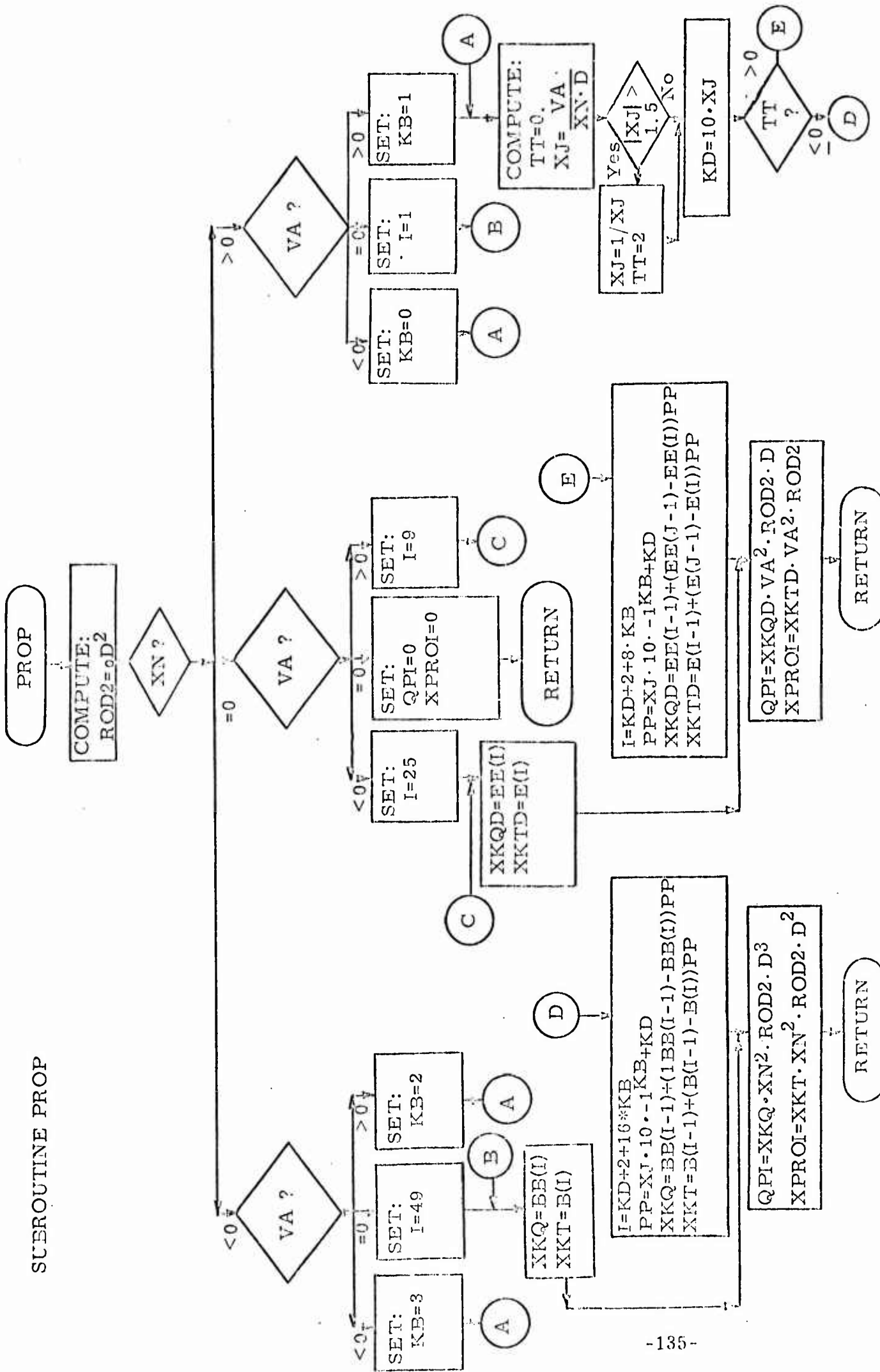
SUBROUTINE QDEV



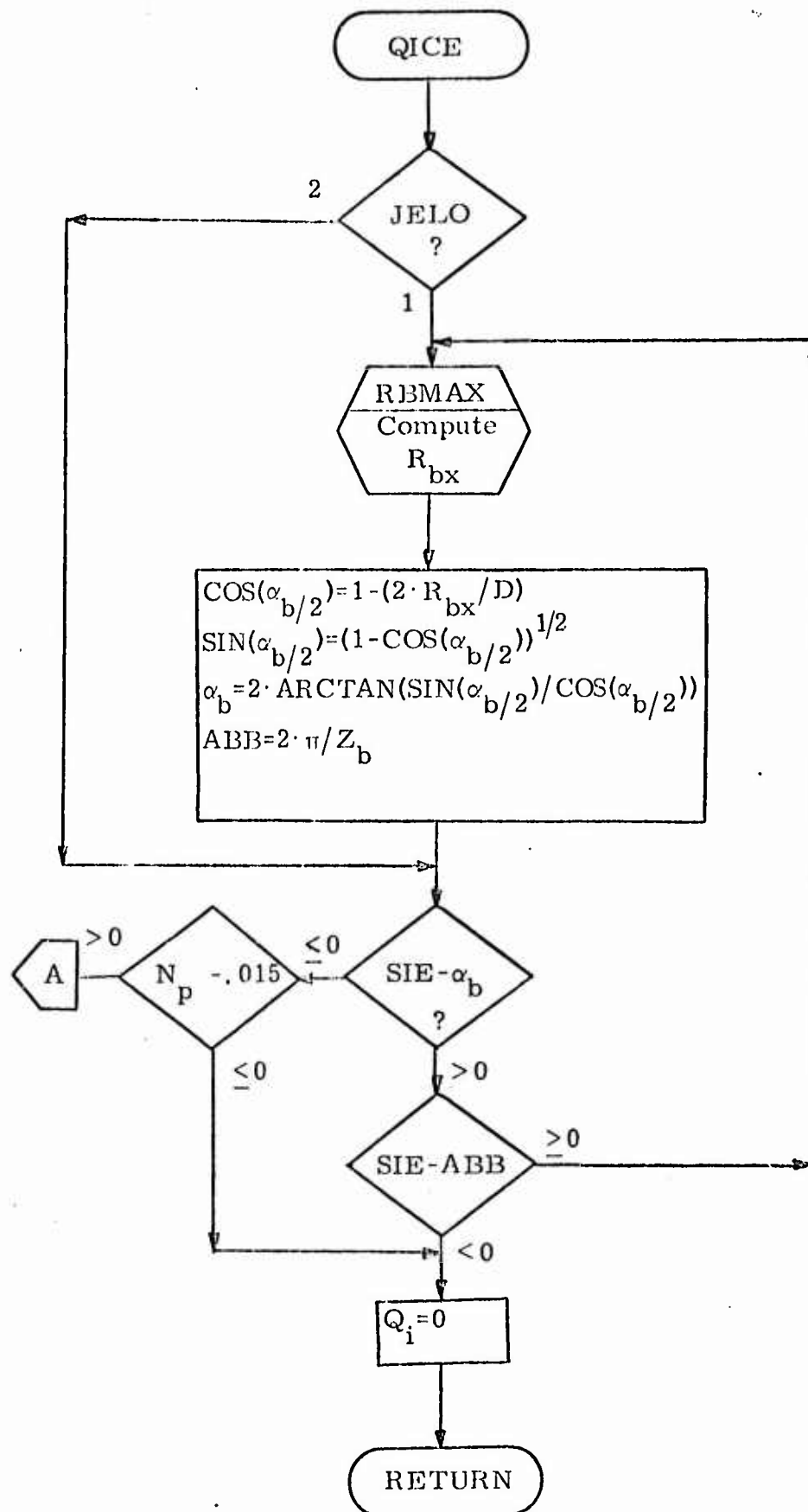
SUBROUTINE HULL.



SUBROUTINE PROP



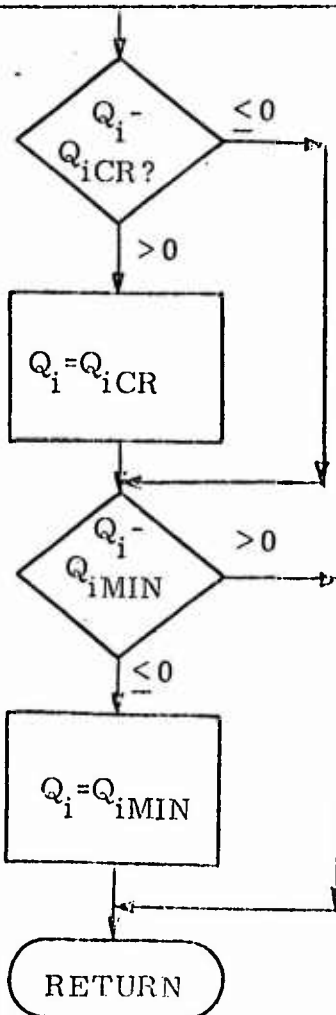
SUBROUTINE QICE



SUBROUTINE QICE (continued)

A

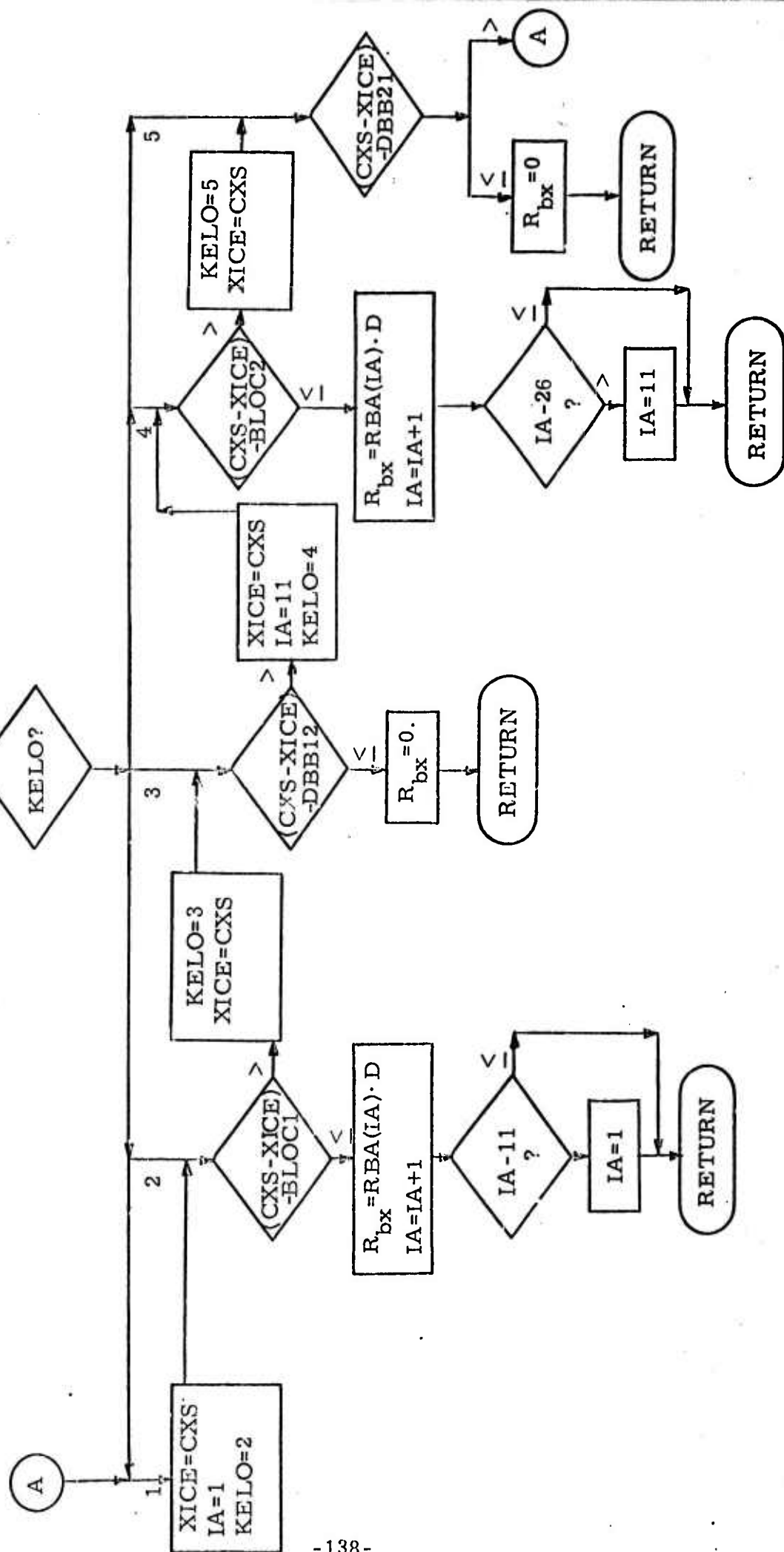
$$\begin{aligned} \cos(\beta) &= \cos(\alpha_{b/2}) \cdot \cos(SIE) + \sin(\alpha_{b/2}) \cdot \sin(SIE) \\ R_b &= D/2(1 - \cos(\alpha_{b/2})/\cos(\beta)) \\ R_m &= (D - R_b)/2 \\ \alpha_v &= \arctan(V_s/(R_m \cdot 2 \cdot \pi \cdot N_p)) \\ \alpha_1 &= (P/D) \cdot D/(2\pi \cdot R_m) \\ e_1 &= e_o + R_b(e_k - e_o)/(D - D_{ct}) \\ B_{avg} &= DAR \cdot \pi \cdot D^2/(2Z_b(D - D_{ct})) \\ Q_{iCRS} &= D_{CR} \cdot e_1 \cdot R_b \cdot R_m + D_{SH} \cdot R_b \cdot R_m \cdot V_s/(Z_b \cdot N_p) \\ Q_{iMIN} &= D_{CR} \cdot E_1 \cdot R_b \cdot R_m \\ Q_{iCR} &= D_{CR} \cdot B_{avg} \cdot R_b \cdot R_m \cdot |\sin(\alpha_1 - \alpha_v)| \\ Q_i &= Q_{iCRS} \end{aligned}$$



SUBROUTINE RBMAX

RBMAX

CSIE=0.
SIE=0.



SUBROUTINE STATE

STATE

$$\begin{aligned}
 ZDO &= \frac{dz}{dt} \\
 \frac{d^2 z}{dt^2} &= 60\alpha_2 \frac{dn_c}{dt} + 60\alpha_2 \alpha_3 n_c - (\alpha_1 + \alpha_d) \frac{dz}{dt} - \alpha_3 \alpha_d z \\
 \frac{dn_{pm}}{dt} &= \frac{Q_{dc} - Q_g}{2\pi J_{pmg}} \\
 VOLT1 &= e_{fg} - (R_{fg} + R_x) i_{fg} \\
 VOLT2 &= e_g - e_c - R_a \cdot i_a \\
 XLD &= L_a + 1.5/XGEN \cdot XLASF \\
 XLM &= \frac{2.25}{XGEN} XM^2 \\
 DENOM &= 1.5 L_{fg} XLD - XLM \\
 \frac{di_a}{dt} &= \frac{1.5(XM \cdot VOLT1 + L_{fg} \cdot VOLT2)}{DENOM} \\
 \frac{di_{fg}}{dt} &= \frac{1.5 XM \cdot VOLT2 + XLD \cdot VOLT1}{DENOM}
 \end{aligned}$$

$$\frac{di_f}{dt} = 0$$

Is $\theta > 0$?

No

Yes

Is $e_b > c_1$?

No

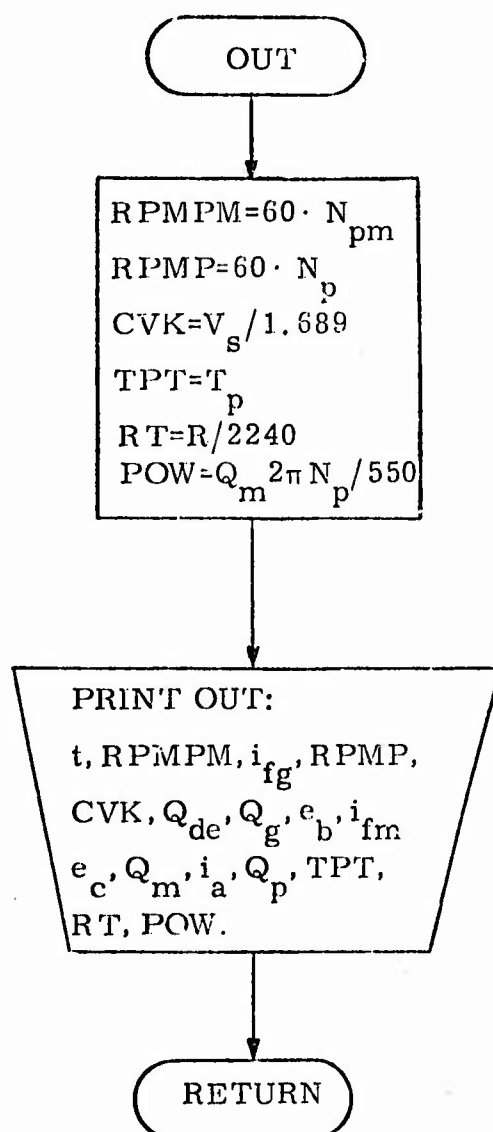
Yes

$$\frac{di_f}{dt} = \frac{.01512(i_a - i_a)}{L_{fm}}$$

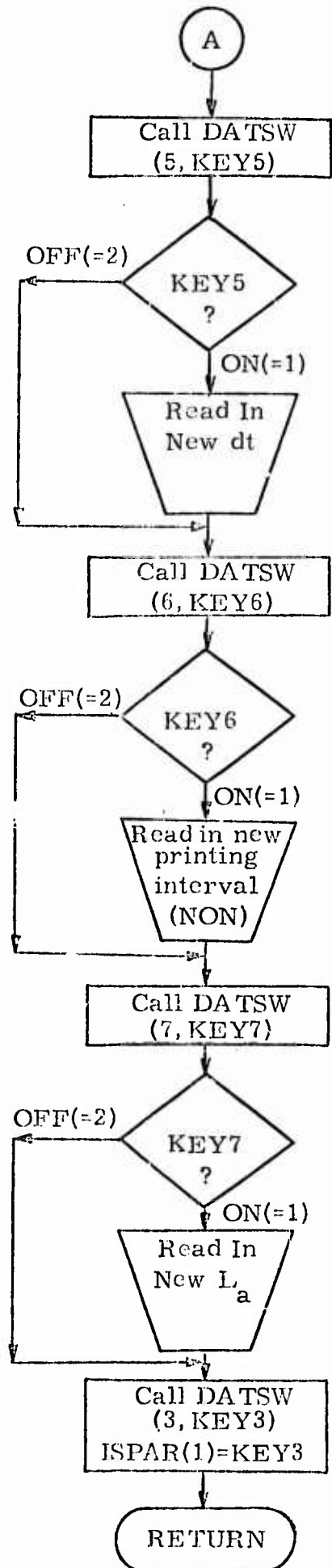
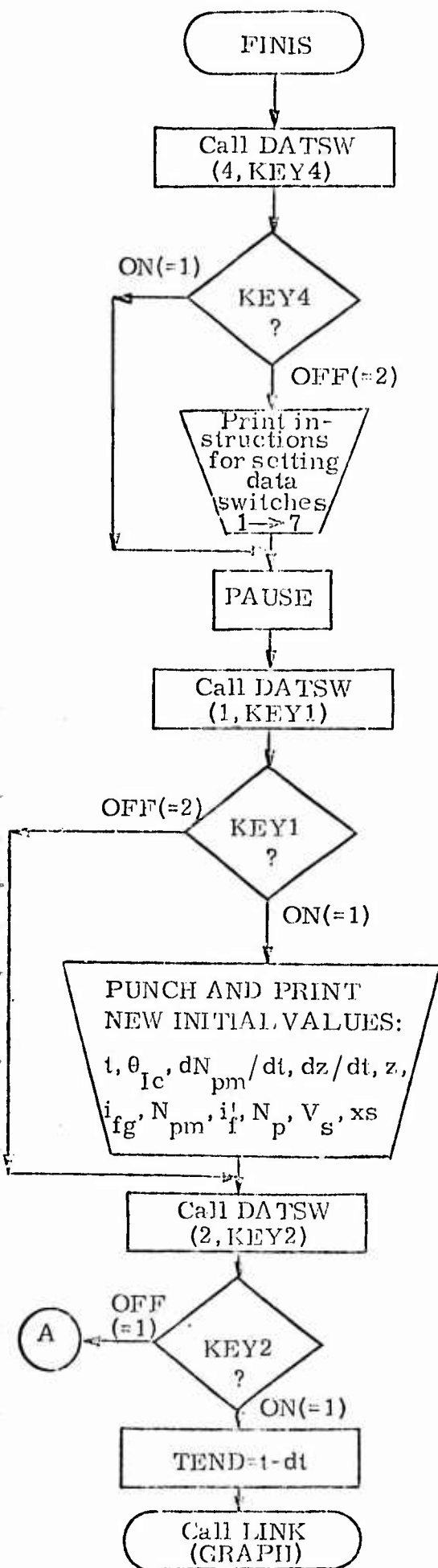
$$\begin{aligned}
 \frac{dn_p}{dt} &= \frac{Q_m - Q_{frm} - Q_p - Q_i}{2\pi J_{mp}} \\
 \frac{dV_s}{dt} &= \frac{T_p - R}{XMASS} \\
 \frac{dX_s}{dt} &= V_s \\
 SIEDO &= N_p \cdot 2\pi
 \end{aligned}$$

RETURN

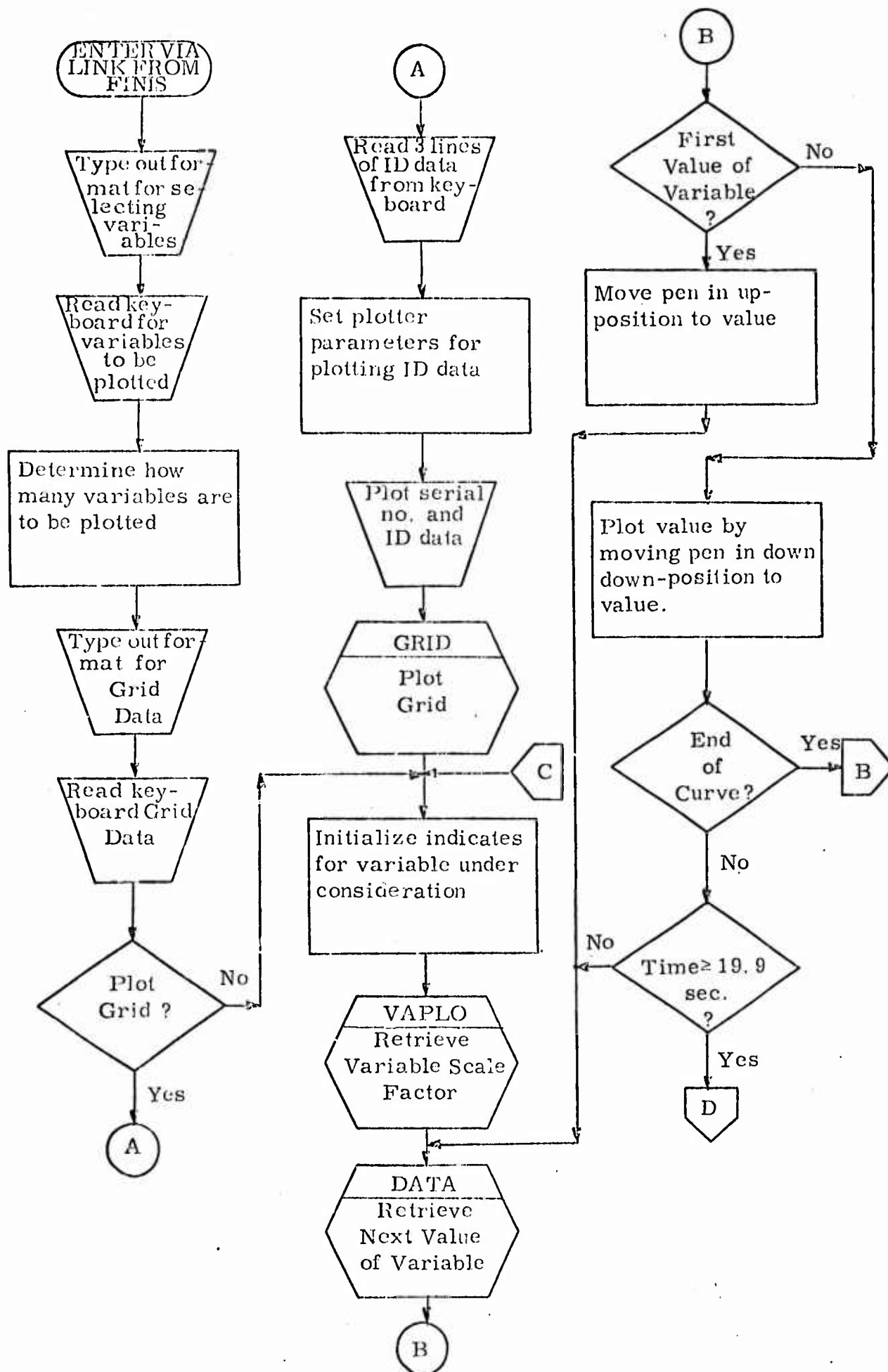
SUBROUTINE OUT



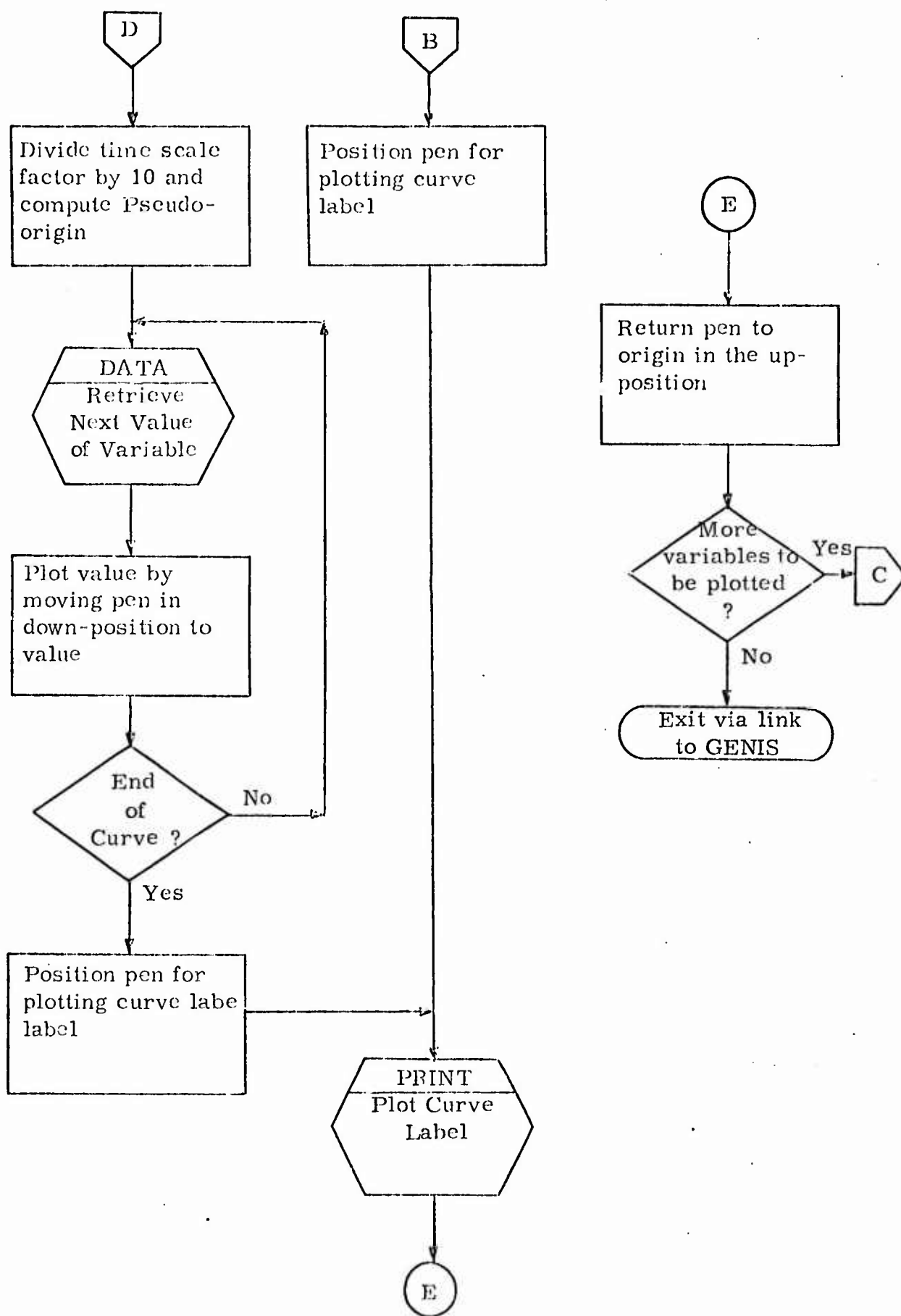
SUBROUTINE FINIS



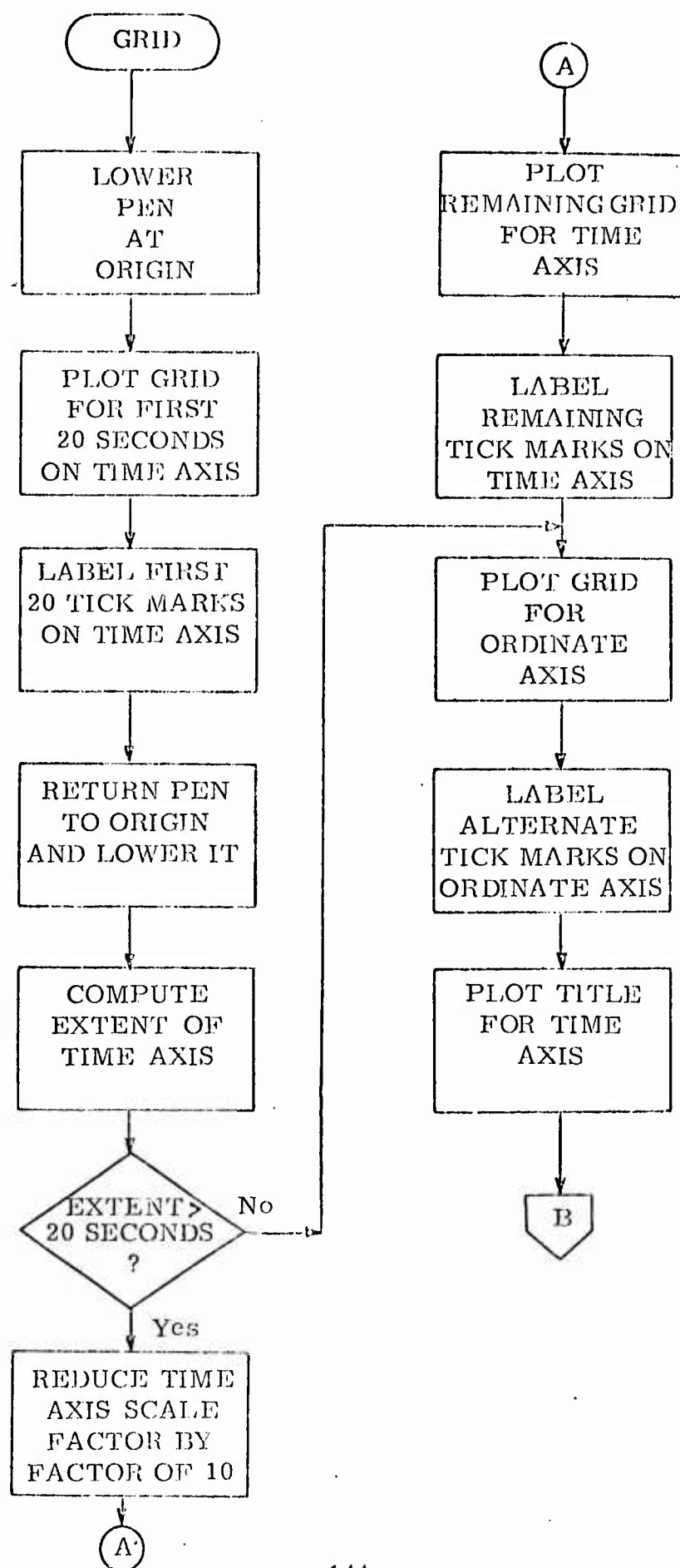
GRAPH MAINLINE



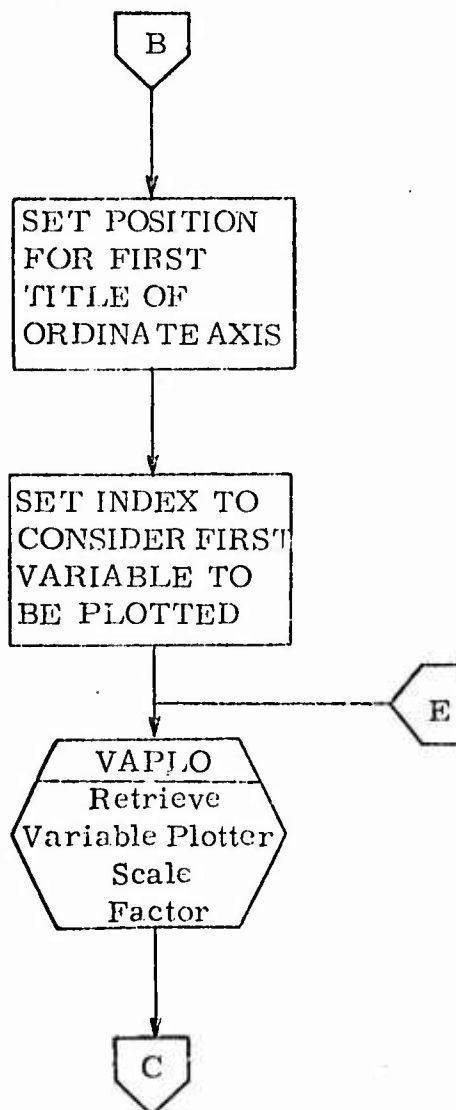
GRAPH MAINLINE (continued)



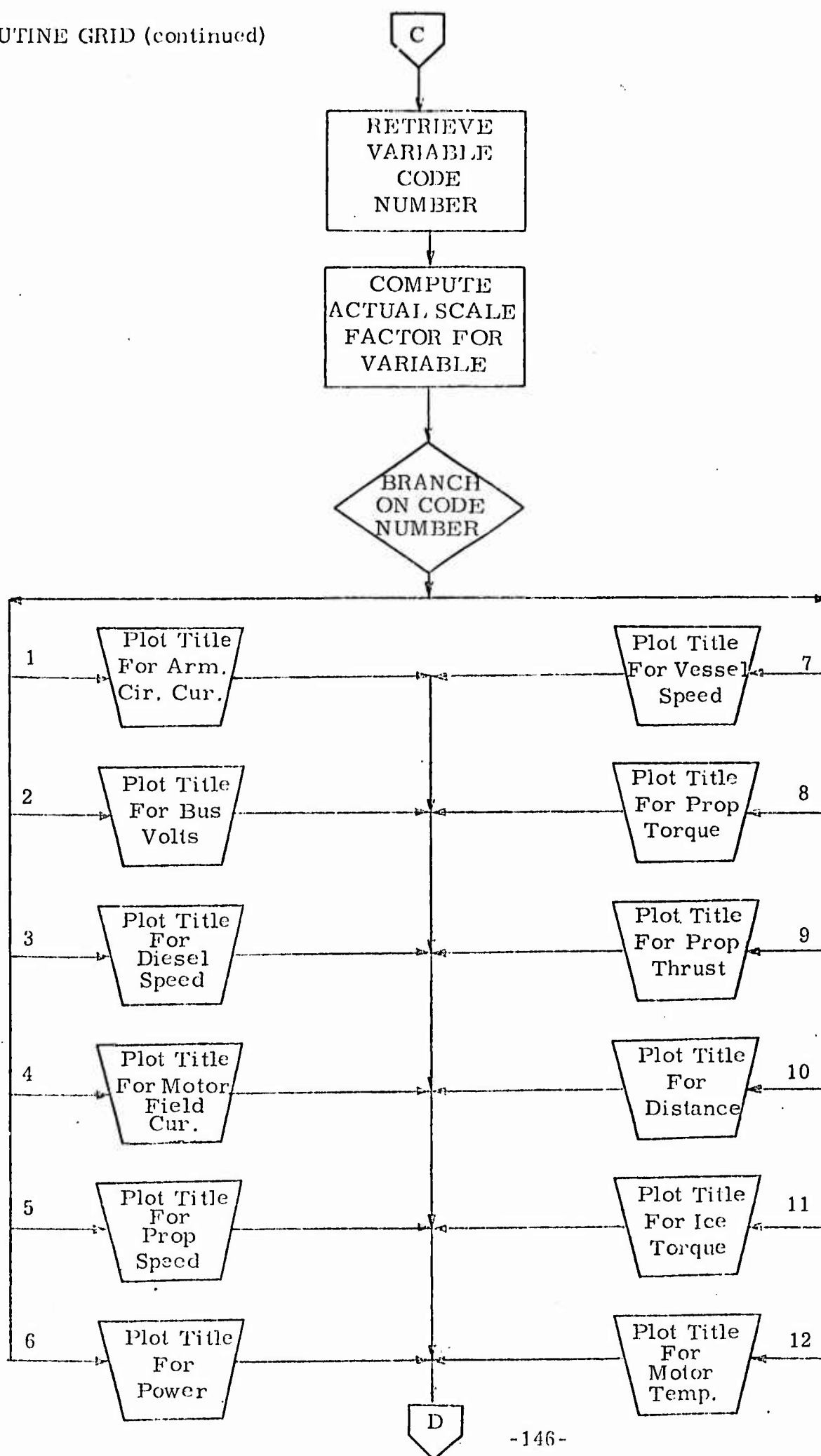
SUBROUTINE GRID



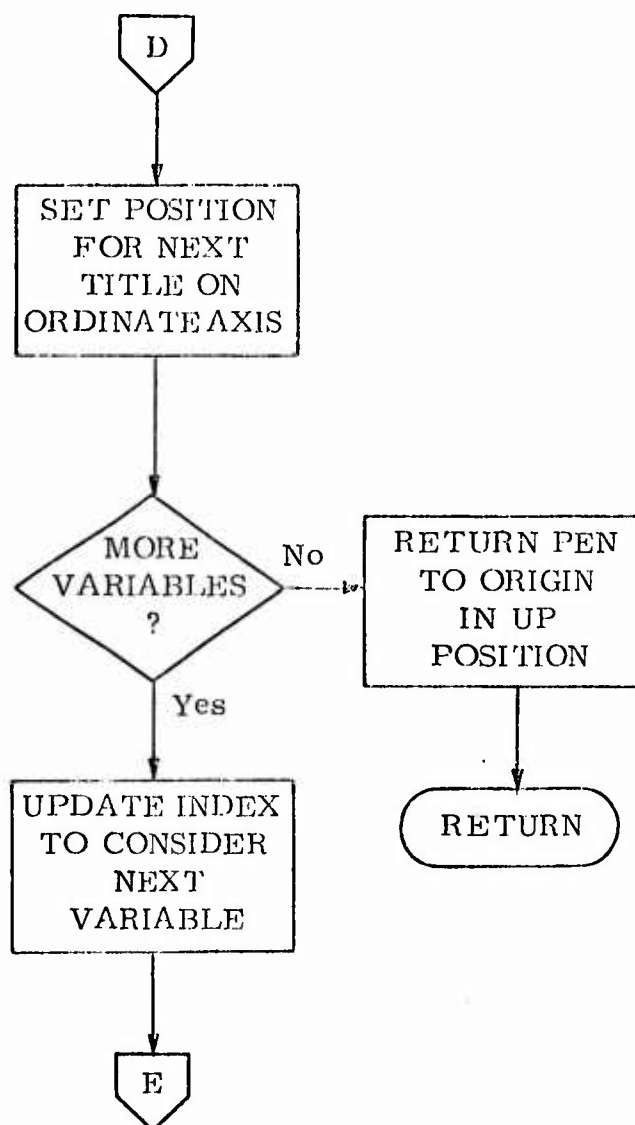
SUBROUTINE GRID (continued)



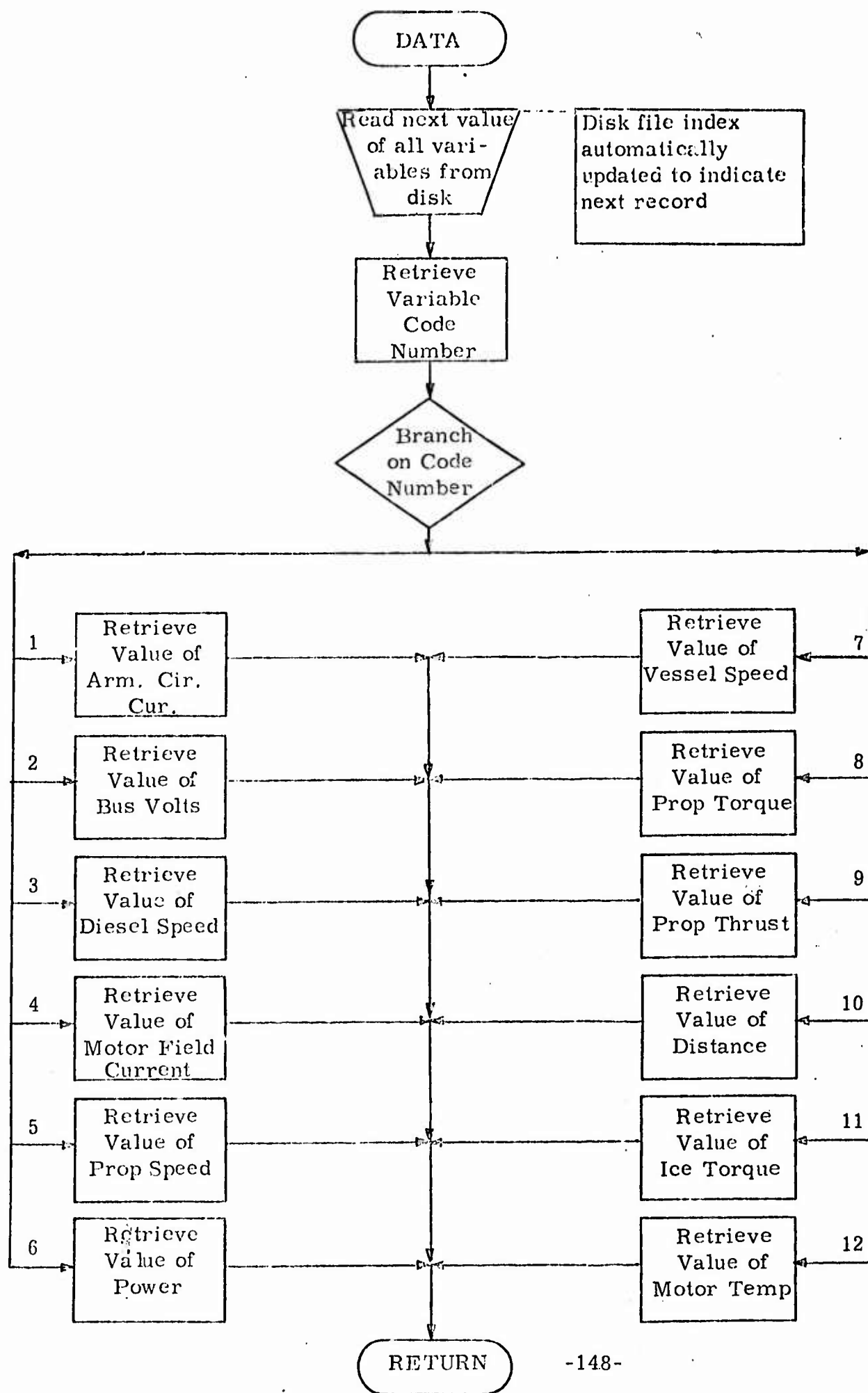
SUBROUTINE GRID (continued)



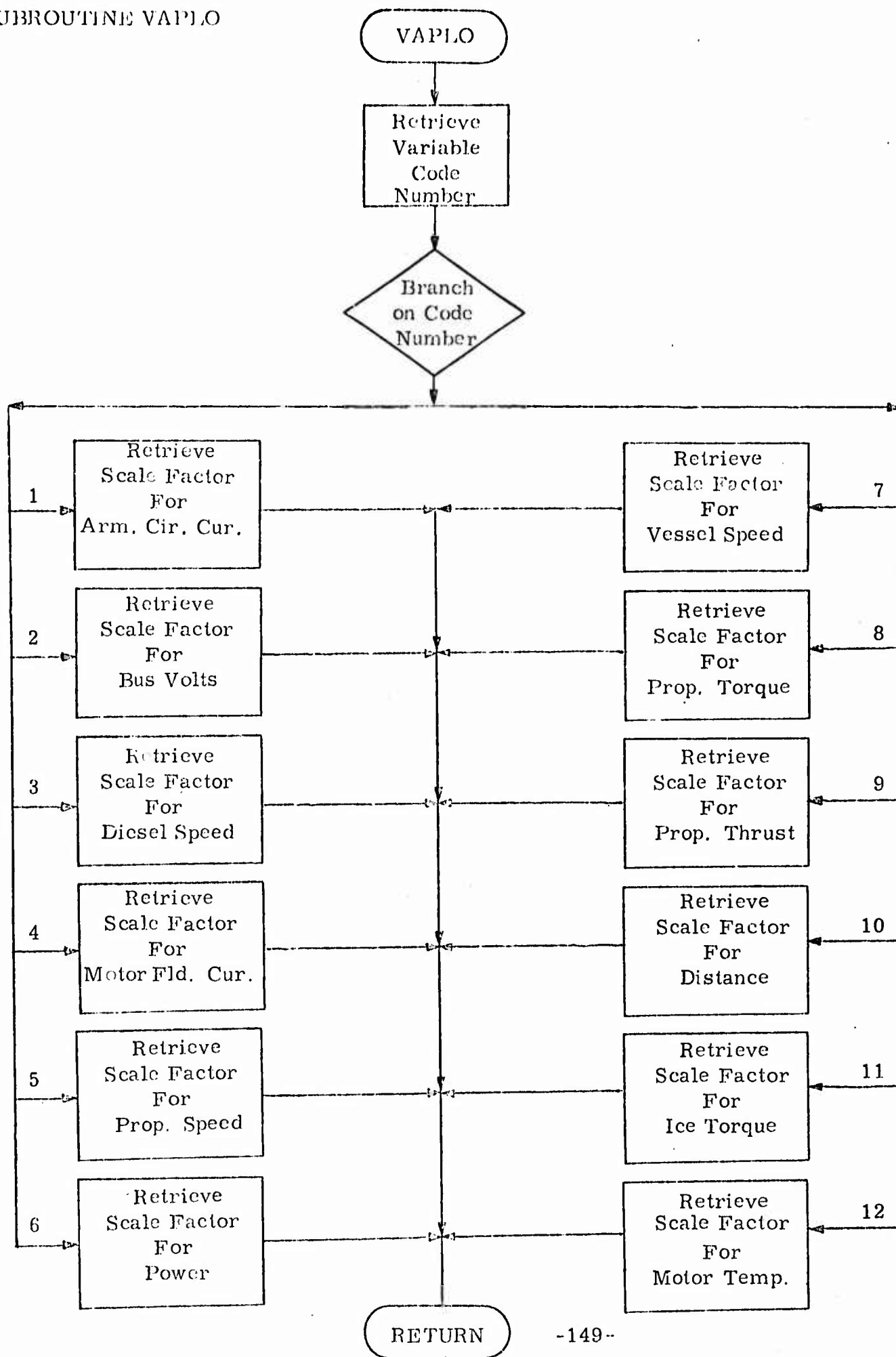
SUBROUTINE GRID (continued)



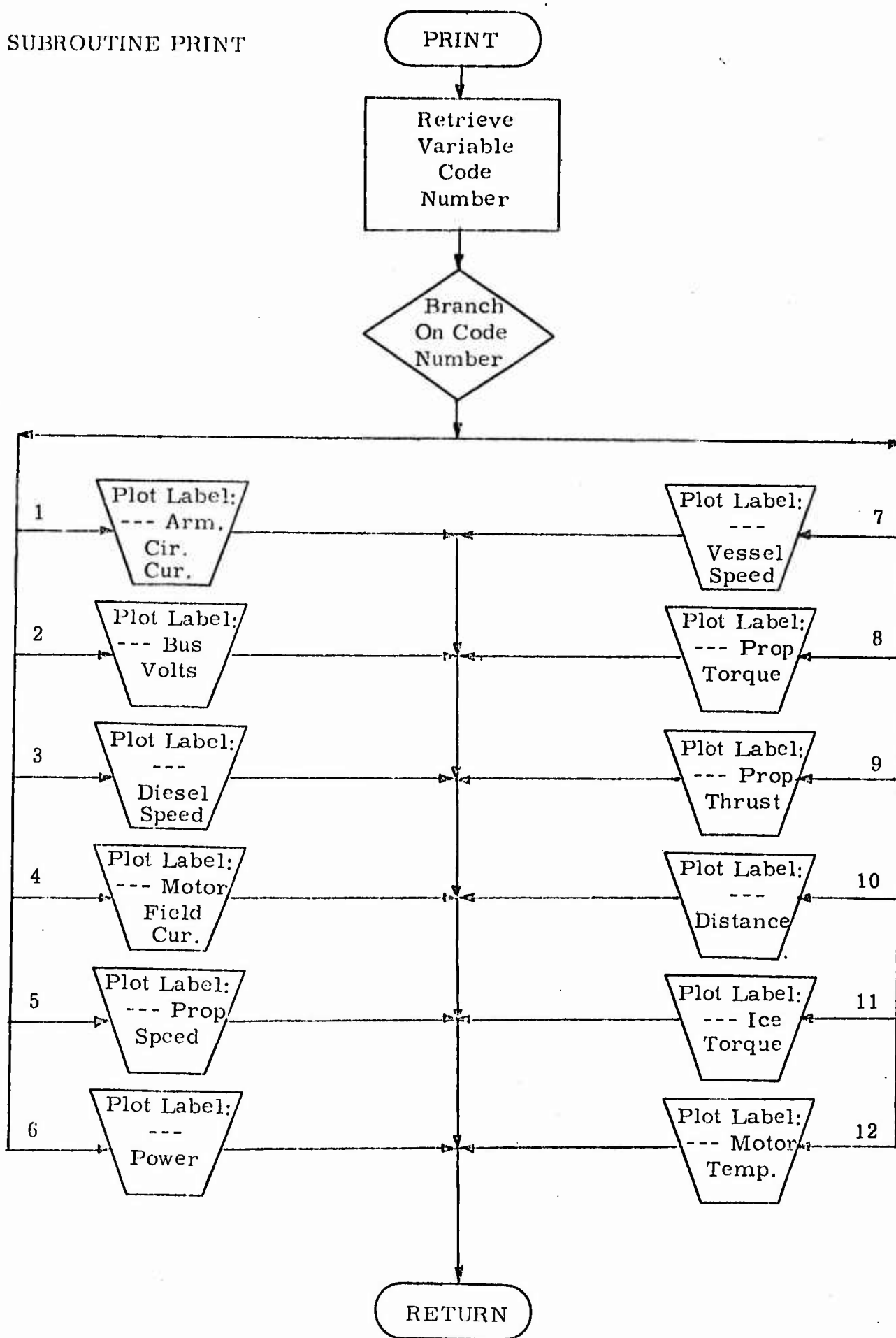
SUBROUTINE DATA



SUBROUTINE VAPLO



SUBROUTINE PRINT



MAINLINE PROGRAMS AND SUBROUTINE

DOCUMENTATION

1. Program Name: Mainline Routine GENIS

2. Program Description: The GENIS mainline routine, after calling subroutine REDAT to read in all the input data and subroutine POLMI to compute the remaining inertias, prints out the control data and initial conditions and depending on the setting of data switches 10 through 15 prints out the ice data, propeller data, hull data, electrical data, governor and bridge controller data, and inertias respectively. The EXEC mainline routine is then loaded and executed (by use of a CALL LINK (EXEC) statement) to begin the simulation.

3. Program Logic: Disk data file 2 contains the serial number of the last simulation. If data switch 9 is on, the serial number will be reset to zero (switch 9 must be on for the first run), otherwise the serial number will be updated by one (1).

4. Listing of Parameters:

Values taken from COMMON - All values read in by subroutine REDAT plus XJPE, the entrained water inertia which was computed in POLMI.

Values taken from the disk - SER, the previous serial number if data switch 9 is off.

Results stored in COMMON- SER, the updated serial number corresponding to this run.

5. Program Length:

- a. Source program - 245 statements (including comments).
- b. Object program - 1828 machine words.

6. Programs Called By This Program: Subroutines REDAT and POLM1; the EXEC mainline is loaded and executed through use of a CALL LINK (EXEC) statement.
7. Programs Calling This Program: This mainline is loaded and executed either by 1) the initial execution as specified on the IBM 1130 Monitor Control Record // XEQ GENIS; 2) the EXEC mainline upon the completion of a run; or 3) the GRAPH mainline upon the completion of a graph.

8. Local Variables:

LIST - 40 word integer array used to store the title
(80 characters or less)

KEY4, KEY9, KEY10, KEY11, KEY12, KEY13, KEY14, and KEY15,
integer variables used to store the setting of the
appropriate data switches; 1 equals on, 2 equals
off.

9. Printed Output: The following lines are printed on the 1132 printer as a title page -

UNITED STATES COAST GUARD ICEBREAKER PROPULSION
SYSTEM SIMULATOR

SIMULATION OF (80 character title)

SERIAL NUMBER xxxxxxxxxxxx

At the top of the next page is printed - SELECTED INPUT DATA
followed by the following headings with their appropriate data-

INITIAL CONDITIONS

CONTROL DATA

PROP DATA
HULL DATA
ELECTRICAL DATA
GOVERNOR AND DIESEL DATA
SYSTEM INERTIAS
ICE DATA

Finally, at the top of a new page is printed OUTPUT DATA.

The following instructions pertaining to the selection of the output data is typed out -

TYPE IN CONCLUSION OF PHRASE 'SIMULATION OF'
USING 80 CHARACTERS OR LESS.

TURN ON APPROPRIATE SWITCHES

ICE DATA -10

PROP DATA -11

HULL DATA -12

ELECTRICAL DATA -13

GOVERNOR AND BRIDGE CONTROLLER DATA -14

INERTIAS -15

TURN ON SELECTOR SWITCHES AND PRESS START.

1. Program Name: SUBROUTINE POLMI
2. Program Description: Subroutine POLMI computes the inertia of the propeller-motor system (J_{mp}) and the combined inertia of the diesel engine and generator armature (J_{pmg}).
3. Program Logic: The number of propeller blades (IZB) determines which of two formulas will be used in computing the "added inertia" of the propeller in water (J_{Pad}). If a value for the propeller inertia in air (J_{Pai}) was not entered in, its value is computed. The values of J_{mp} and J_{pmg} are then calculated.
4. Listing of Parameters: All parameters are in COMMON.
Values taken from COMMON - IZB, XJPA, XJM, XJSH, XJD, XJG, XGEN, and D.
Results stored in COMMON - XJPA, XJMP, XJDG, and XJPE.
5. Program Length:
 - a. Source program - 45 statements (including comments).
 - b. Object program - 112 machine words.
6. Programs Called By This Program: None
7. Programs Calling This Program: GENIS mainline.
8. Local Variables: None
9. Printed Output: None

1. Program Name: SUBROUTINE REDAT
2. Program Description: Subroutine REDAT reads in all 104 data cards that represent the propeller data, hull data, electrical data, governor-diesel data, systems inertias, ice conditions, scale factors, initial conditions and control data.
3. Program Logic: After reading in all the data cards, the program sets the values of ρ and π to 1.99 and 3.14159 respectively. The armature circuit resistance is calculated from the resistance of the motor armature (R_M), the resistance of the generator armature (R_G), and the number of generators (XGEN). The "mass" term is also calculated.
4. Listing of Parameters: All parameters are located in COMMON. Values taken from COMMON- none.
Results stored in COMMON - the variables initially read in by this program are so indicated under the COMMON data description section.
5. Program Length:
 - a. Source program - 127 statements (including comments).
 - b. Object program - 498 machine words.
6. Programs Called By This Program: None
7. Programs Calling This Program: GENIS mainline.
8. Local Variables: None
9. Printed Output: None

1. Program Name: Mainline Routine EXEC
2. Program Description: The EXEC mainline routine coordinates the actual simulation. Subroutine CNTRL is called to compute the initial values after which subroutine OUT is called to print them out. Thereafter subroutine HEUN is called each time a new time step is taken. Subroutine OUT is called each time the values are to be printed out. Upon conclusion of the simulation subroutine FINIS is called to select control for the remainder of the run. If FINIS returns (implies a graph was not desired) then the GENIS mainline may be loaded and executed if it is desired to start a new problem, otherwise HEUN is called to continue the present problem. The EXEC mainline writes the values on the disk for use by the GRAPH mainline.
3. Program Logic: If data switch zero (0) is on after the initial conditions are computed subroutine FINIS will be called, otherwise it will not be. The variable NON controls the ship interval for printing with the variable MIKE being a counter of the steps taken.
4. Listing of Parameters: All parameters are in COMMON .
Values taken from COMMON - TIMSC, NON, CT, TEND, XNPM, XNP, CVS, QM, PI, AI, EB, FMI, CXS, QPI, XPROI, IELO, QI, and ISPAR(1).
Results stored in COMMON - IELO, JELO, KELO, and ISPAR(2).
5. Program Length:
 - a. Source program - 108 statements (including comments).
 - b. Object program - 264 machine words.

6. Programs Called By This Program: Subroutines CNTRL, OUT, HEUN, and FINIS. The GENIS mainline may be loaded and executed through use of a CALL LINK (GENIS) statement.
7. Programs Calling This Program: Loaded and executed by the GENIS mainline through use of a CALL LINK(EXEC) statement.
8. Local Variables:

III - initially set to one (1) but will take on a value of two(2) the first time the value of the current time (CT) exceeds 20 seconds; is used to reset the disk write time step (DTT) to 1.0 after CT exceeds 20 seconds.

MIKE - initially set to one (1) but is updated by one each time a new time step is completed; is used to count time steps for comparison with print out time step (NON).

DTT - disk write time step; is mutually set to the value of TIMSC but will take on the value of one (1) after CT exceeds 20.0 seconds.

TTT - time value for next writing on disk; is updated by DDT each time the values are written on the disk.

KEYO - contains setting of data switch zero; equals one (1) if on, two (2) if off.

RPMPM- angular velocity of prime mover in revolutions per minute.

RPMP - propeller angular velocity in revolutions per
minute.

CVK - ship's speed in knots

XKW - ship's horsepower

JENE - set to value of ISPAR(1) for use in computed
GO TO Statement.

9. Printed Output: None

1. Program Name: SUBROUTINE CNTRL
2. Program Description: Subroutine CNTRL sets the value of the ordered bridge lever position (θ_o) depending on the value of ICNTR which indicates the command that was given. Subroutine CALVR is then called to calculate the initial values, after which the current values are saved.
3. Program Logic: The value of θ_o is set to 165.0° if ahead flank was given (ICNTR=1), 135.0° if either ahead full (ICNTR=3) or ahead full in ice (ICNTR=4) was given, and -165.0° if back flank (ICNTR=2) was given. The current values of the eleven state variables ($t, i_a, dz/dt, z, n_{pm}, i'_f, i_{fg}, n_p, V_s, XS$, and SIE) are placed into alternate locations. The initial values are then written on the disk and a heading is printed out to label the columns for succeeding print outs.
4. Listing of Parameters: All parameters are in COMMON.

Values taken from COMMON - ICNTR, FIMAX, FIP, T, AI, Y, Z, XNPM, FIP, FGI, XNP, VS, XS, QM, PI, EB, QPI, XPROI, QI and IELO.

Results stored in COMMON - SIE, THETO, FMI, CT, CAI, CY, CZ, CXNPM, CFIP, CGI, CXNP, CUS, CXS, CSIE, and IELO.
5. Program Length:
 - a. Source program - 84 statements (including comments).
 - b. Object program - 242 machine words.
6. Programs Called By This Program: Subroutine CALVR

7. Programs Calling This Program: EXEC mainline

8. Local Variables:

RPMPM- angular velocity of prime mover in revolutions
per minute.

RPMP - propeller angular velocity in revolutions per
minute.

CVK - ship's speed in knots

XKW - ship's horsepower

9. Printed Output: Prints out the column headings for the output
data that appears in the printer as follows:

T(SEC) RPMPM CFGI RPMP VK QDE(FT-LB)

QG EB(V) FMI EC QM(FT-LB) AI(A)

QP THRUST(T) RESIST HP /SH

1. Program Name: SUBROUTINE HEUN

2. Program Description: Subroutine HEUN is the numerical integration routine. The Runge-Kutta 4th order method is used. Very briefly, the method is as follows: we are seeking to solve the equation $dy/dx = f(x, y)$. Given an (x_n, y_n) , we calculate: $y_{n+1} = y_n + (k_1 + 2k_2 + 2k_3 + k_4)/6$
 where: $k_1 = hf(x_n, y_n)$
 $k_2 = hf(x_n + h/2, y_n + k_1/2)$
 $k_3 = hf(x_n + h/2, y_n + k_2/2)$
 $k_4 = hf(x_n + h, y_n + k_3)$
 and h = independent axis increment (dx).

3. Program Logic: Before subroutine CALVR is called each time, a check is made on the values of Z and FIP. If Z is greater than 1.000 Z is set to 1.000 if FIP is negative, FIP is set to zero. The local variable IJAX is used to return control to the point in the routine where the branch was taken to check on the values of Z and FIP.

4. Listing of Parameters: All parameters are in COMMON. Values taken from and stored in COMMON - the values and derivatives are represented by the three arrays STVR, CSTVR, and STVRD. The array STVRD corresponds to Z through TEMP in COMMON, CSTVR corresponds to CZ through CTEMP, and STVRD corresponds to ZDO through TEMDO. T and DT are also used.

1. Program Name: SUBROUTINE HEUN

2. Program Description: Subroutine HEUN is the numerical integration routine. The Runge-Kutta 4th order method is used. Very briefly, the method is as follows: we are seeking to solve the equation $dy/dx = f(x, y)$. Given an (x_n, y_n) , we calculate: $y_{n+1} = y_n + (k_1 + 2k_2 + 2k_3 + k_4)/6$

where: $k_1 = hf(x_n, y_n)$

$$k_2 = hf(x_n + h/2, y_n + k_1/2)$$

$$k_3 = hf(x_n + h/2, y_n + k_2/2)$$

$$k_4 = hf(x_n + h, y_n + k_3)$$

and h = independent axis increment (dx).

3. Program Logic: Before subroutine CALVR is called each time, a check is made on the values of Z and FIP. If Z is greater than .000 Z is set to 1.000 if FIP is negative, FIP is set to zero. The local variable IJAX is used to return control to the point in the routine where the branch was taken to check on the values of Z and FIP.

4. Listing of Parameters: All parameters are in COMMON. Values taken from and stored in COMMON - the values and derivatives are represented by the three arrays STVR, CSTVR, and STVRD. The array STVRD corresponds to Z through TEMP in COMMON, CSTVR corresponds to CZ through CTEMP, and STVRD corresponds to ZDO through TEMDO. T and DT are also used.

5. Program Length:

- a. Source program - 84 statements (including comments).
- b. Object program - 354 machine words.

6. Programs Called By This Program: Subroutine CALVR.

7. Programs Calling This Program: EXEC mainline.

8. Local Variables:

IJAX - takes on the values one (1) through 4 to control the 4 times subroutine CALVR is called.

DD - set to $DT/2$ for use in solving for k_2 and k_3 .

9. Printed Output: None

1. Program Name: SUBROUTINE CALVR
2. Program Description: Subroutine CALVR updates the values of all the working variables in the program. This is accomplished by calling the following subroutines: CON1, CON2, QDEV, FLUXG, FLUXM, HULL, PROP, QICE (if the ship has encountered ice), and STATE. Intermixed between calls to the above subroutines, subroutine CALVR does some calculations and adjusting of its own. This subroutine is called by subroutine HEUN (the numerical integration routine) each time an updated set of values is desired.
3. Program Logic: Subroutine QICE is called only after the ship has encountered ice which is controlled by the variable TICE - the relative time since the beginning of the run before the ship encounters ice. The variable IPLUG (equivalenced to ISPAR(2)) is initially set to one (1) so that the initial values of T and FMI will be saved for use in the calculation of the updated values of FMI. Thereafter IPLUG is set to two(2) to immediately calculate a new value of FMI using the saved initial values.
4. Listing of Parameters: All parameters are in COMMON. Values taken from COMMON-XNGR, NXPM, NXGRD, XNPMD, QDE, QBRG, CWM, XNPM, XGEN, TRNFG, FGI, TRNSF, AI, CFG, CFM, XNP, RG, IPLUG, T, FMI, TAU A, TAUB, FMIRV, THETA, C1, XK1, FIMAX, FIP, CSHFR, EB, WDED, VS, QPI, TDED, XPROI, XKK, CT, and TICE.
Results stored in COMMON - XNE, XNEDO, QDE, AMPTR, EG, EC, QG, EB, QM, FMI, AIHAT, QFRM, VA, QP, TP, and QI.

5. Program Length:

- a. Source program - 110 statements (including comments).
- b. Object program - 310 machine words.

6. Programs Called By This Program: Subroutines CON1, CON2, QDEV, FLUXG, FLUXM, HULL, PROP, QICE, and STATE.

7. Programs Calling This Program: Subroutines HEUN and CNTRL.

8. Local Variables:

TTN - contains initial value of T.

FMIC - contains initial value of FMI

ABLE, BAKER - set to exponential expressions used in the calculation of FMI.

9. Printed Output: None

1. Program Name: SUBROUTINE CON1 .
2. Program Description: Subroutine CON1 computes the actual bridge lever position (θ) and bridge lever time rate ($d\theta/dt$) which depends on time (t) and control values τ_1 and τ_2 .
3. Program Logic: The control values τ_1 and τ_2 determine the speed of the lever in moving from the initial position (θ_{IC}) to the final or ordered position (θ^*). The lever moves from θ_{IC} to θ_1 in τ_1 seconds and from θ_1 to θ^* in τ_2 seconds. A check is made to determine where the present value of t lies, after-which θ and $d\theta/dt$ are computed accordingly.
4. Listing of Parameters: All parameters are located in COMMON. Values taken from COMMON - T, TAU1, TAU2, THET0, THET1, and THEIC
Results stored in COMMON - THETA and THETD.
5. Program Length:
 - a. Source program - 45 statements (including comments).
 - b. Object program - 82 machine words.
6. Programs Called By This Program: None
7. Programs Calling This Program: Subroutine CALVR.
8. Local Variables: None
9. Printed Output: None

1. Program Name: SUBROUTINE CON2
2. Program Description: Subroutine CON2 computes the ordered generator field voltage (e_{fg}), ordered diesel speed (n_{gr}) and ordered diesel speed rate (dn_{gr}/dt). The values calculated are dependent on the values of the actual bridge lever position (θ) and bridge lever time rate ($d\theta/dt$) computed in subroutine CON1.
5. Program Logic: A maximum value of e_{fg} is calculated from the generator field resistance (R_{fg}) and variable resistor in the generator field (R_x). If the value of θ is below 63.0 degrees, the diesel is still idling and the value of dn_{gr}/dt will be zero. If θ is greater than 63.0 degrees, a check is made to determine if a full ($\theta=135.0$ degrees) or maximum ($\theta=165.0$ degrees) command was given. The sign of e_{fg} is determined to be negative if θ is negative and positive if θ is positive.
4. Listing of Parameters: All parameters are located in COMMON. Values taken from COMMON RX, RFG, THETA, THETD and THETD. Results stored in COMMON - EFG, XNGR, and XNGRD.
5. Program Length:
 - a. Source program - 56 statements (including comments).
 - b. Object program - 178 machine words.
6. Programs Called By This Program: None
7. Programs Calling This Program: Subroutine CALVR.
8. Local Variables:

EFGM - real variable containing the computed maximum
value of e_{fg} .

THET - real value containing the absolute value of θ .

9. Printed Output: None.

1. Program Name: SUBROUTINE FLUXG
2. Program Description: Subroutine FLUXG calculates the generator flux (C_{fg}), generator field inductance (L_{fg}), generator time constant (τ_{fg}), mutual inductance between generator separate and in series fields (XM), and the inductance of the generator in series field (XLASF) from the value of ampere-turns (AMPTR), a table of field flux (CFMA), values of control field turns (TRNFG), series field turns (TRNSF), field resistances (RX, RG) and flux conversion constant (ZGP).
3. Program Logic: The generator field flux is taken from the input array (CFGA) with the value of the generator spacing (SPGEN) being the independent axis. Since the independent axis runs from zero to 12 it is necessary to truncate the independent axis index (KG) and then compute the enclosing indices by adding two and one to the value.
4. Listing of Parameters: All parameters are in COMMON.
Values taken from COMMON-AMPTR, SPGEN, CFGA, TRNFG
ZGP, RX, RG, and TRNSF.
Results stored in COMMON-CFG, XLFG, TAUFG, XM, and XLASF.
5. Program Length:
 - a. Source program - 52 statements (including comments).
 - b. Object program - 148 machine words.
6. Programs Called By This Program: None
7. Programs Calling This Program: Subroutine: CALVR
8. Local Variables:

FG-	real variable containing the actual value of the independent axis index.
-----	--

KG - integer variable containing the truncated value
of FG.
DCGDI - real variable containing the interval that the value
of FG falls in.

9. Printed Output: None

1. Program Name: SUBROUTINE FLUXM
2. Program Description: Subroutine FLUXM calculates the motor field flux (C_{fm}), the motor field time constant (τ_{fm}), and the motor field inductance (L_{fm}) from the value of field current (FMI), a table of field flux (CFMA), values of field turns (TRNFM), field resistance (RFM) and flux conversion constant (ZMP).
3. Program Logic: The motor field flux is taken from the input array (CFMA) with the value of the motor spacing (SPMOT) being the independent axis. Since the independent axis runs from zero to 12, it is necessary to truncate the independent axis index (KM) and then compute the enclosing indices by adding two and one to the value.
4. Listing of Parameters: All parameters are in COMMON.
Values taken from COMMON-FMI, SPMOT, DFMA, TRNFM, ZMP, and RFM.
Results stored in COMMON-DFM, XLFM, and TAUFM.
5. Program Length:
 - a. Source program - 52 statements (including comments).
 - b. Object program - 112 machine words.
6. Programs Called By This Program: None
7. Programs Calling This Program: Subroutine CALVR.
8. Local Variables:

FM - real variable containing actual value of independent axis index.

KM- integer value containing truncated value of FM
DCMDI - real value containing the interval that the value of
FM falls in.

9. Printed Output: None

1. Program Name: SUBROUTINE QDEV
2. Program Description: Subroutine QDEV calculates the developed diesel torque (Q_{de}) using the developed torque constant (K_6), the servo stroke position (Z), the number of generators (XGEN), and the angular velocity of the prime mover (N_{pm}).
3. Program Logic: The maximum developed diesel torque ($Q_{de_{max}}$) is computed from the angular velocity of the prime mover (N_{pm}). A diesel engine friction torque is computed using a friction torque - N_{pm} curve with a Y-intercept indicated by the value of QBRG and a slope indicated by CWM. This value is added to the old value of $Q_{de_{max}}$ to arrive at an augmented $Q_{de_{max}}$. If the value of Q_{de} is greater than $Q_{de_{max}}$, Q_{de} is set equal to $Q_{de_{max}}$.
4. Listing of Parameters: All parameters are in COMMON. Values taken from COMMON - XK6, Z, XNPM, XGEN, QBRG, and CWM. Results stored in COMMON - QDE
5. Program Length:
 - a. Source program - 53 statements (including comments).
 - b. Object program - 138 machine words.
6. Programs Called By This Program : None
7. Programs Calling This Program : Subroutine CALVR
8. Local Variables:

$QDEM_X$ - real variable containing the maximum developed diesel torque.
9. Printed Output: None

1. Program Name: SUBROUTINE HULL
2. Program Description: Subroutine HULL computes the ship's resistance (R), wake deduction (WDED), and thrust deduction (TDED) from the ship's speed (V_S), ice resistance factor (CICE) and input tables of effective horsepower (EHP), wake deduction (WADED) and thrust deduction (THDED).

3. Program Logic: If the control value ICNTR implies ahead full in ice (equal to 4), the ice resistance factor (CICE) is taken into account in computing the resistance of the hull. If the ship's speed is negative, the hull resistance is taken as 1.4 times the computed value and the wake and thrust deductions are taken as 1.0 and 0.9 respectively.

Since the independent axis of the three tables (EHP, WADED, and THDED) runs from zero to 19.0, it is necessary to add two and one to the truncated index (KV) in order to arrive at the interval the actual index (V) falls between.

4. Listing of Parameters: All parameters are in COMMON.
Values taken from COMMON-VS, EHP, WADED, THDED, ICNTR, and CICE.
Results stored in COMMON - WDED, TDED, and R.

5. Program Length:
 - a. Source program - 52 statements (including comments).
 - b. Object program - 152 machine words.

6. Programs Called By This Program: None

7. Programs Calling This Program: Subroutine CA1.VR

8. Local Variables:

V -	real variable containing the absolute value of the ship's speed in knots.
KV -	integer variable containing truncated value of V.
REHP -	real variable containing the effective horsepower used in calculating R.

9. Printed Output: None

1. Program Name: SUBROUTINE PROP
2. Program Description: Subroutine PROP makes use of the propeller data tables to compute the ideal propeller torque (QPI) and thrust per propeller (XPROI) given the propeller angular velocity (n_p) advance velocity of the propeller (Va).
3. Program Logic: The values of the propeller thrust and torque coefficients are computed from the tables by a complex but efficient means. Depending on the signs of VA and XN, a value of the variable KB is set (see Local Variables). This value is used in computing the table subscript and also to set the sign of the interpolation fraction (PP) to the negative. A value of XJ or 1/XJ is computed. The integer variable KD is set to the truncated value of 10 times $|XJ|$. This will determine the relative position of I in its quadrant. The actual value of I in the table is computed using KD and KB. The interpolation fraction (PP) is computed and its sign adjusted by using KB to be negative. The coefficients can now be computed from the tables.
4. Listing of Parameters: All parameters are in COMMON. Values taken from COMMON - RO, D, XN, VA, B, BB, E, and EE. Results stored in COMMON - QPI, and XPROI.
5. Program Length:
 - a. Source program - 106 statements (including comments).
 - b. Object program - 430 machine words.

6. Programs Called By This Program: None

7. Programs Calling This Program: Subroutine CALVR

8. Local Variables:

KB - takes on the values zero (0) through 3 depending on the signs of XN and VA as follows.

KB	0	1	2	3
XN	+	+	-	-
VA	-	-	+	-

I - index to data arrays (B, BB, E, and EE).

TT - equal equal to zero (0) if using J, 1 if using 1/J

XJ - equal to $VA/(XN \cdot D)$ if result within range $/XJ/ \leq 1.5$, otherwise equal to $XN \cdot D/VA$

KD - equal to $10 \cdot /XJ/$ and used in computing I

PP - negative fraction used in interpolation

XKQ - propeller torque coefficient

XKT - propeller thrust coefficient

XKQD - modified propeller torque coefficient

XKTD - modified propeller thrust coefficient

9. Printed Output: None

1. Program Name: SUBROUTINE QICE
2. Program Description: Subroutine QICE computes the ice torque (Q_i) of the propeller. If necessary, subroutine RBMAX is called to determine a new value of the maximum depth of cut in feet that the propeller will make in the ice. The value of Q_i is taken as the lesser of crushing torque (Q_{iCR}) and combined crushing and shearing torque (Q_{iCRS}) except that Q_i will never be less than the minimum ice torque (Q_{iMIN}).
3. Program Logic: The variable JELO is initially set to one (1) in the EXEC mainline so that subroutine RBMAX will be called immediately the first time this subroutine is executed. Thereafter, subroutine RBMAX is called only when a new value of the depth of cut is needed (when the angular position of the propeller is greater than the cut angle, α_b , and greater than or equal to the angle between propeller blades.
4. Listing of Parameters: All parameters are located in COMMON. Values taken from COMMON-JELO, RBMX, D, IZB, PI, SIE, XNP, PDR, DAR, DCR, EO, EK, DCT, DSH, and VS. Results stored in COMMON-JELO and QI.
5. Program Length:
 - a. Source program - 75 statements (including comments).
 - b. Object program - 352 machine words.
6. Programs Called By This Program: Subroutine RBMAX
7. Programs Calling This Program: Subroutine CALVR

8. Local Variables:

CSAB2 - real variable containing the cosine of $\alpha_b/2$
SNAB2 - real variable containing the sine of $\alpha_b/2$
AB2 - real variable containing the value of α_b
XIZB - real variable containing the number of propeller blades.

ABB - real variable containing the number of radians between prop blades.

CSBET - real variable containing the cosine of θ
RB - real variable containing value of R_b
RM - although appearing in COMMON, this local real variable contains the value of R_m

ALFAV - real variable containing value of α_v
ALFA1 - real variable containing value of α_1
E1 - real variable containing value of e_1
BAV - real variable containing value of B_{avg}
QICRS - real variable containing value of Q_{iCRS}
QIMIN - real variable containing value of Q_{iMIN}
QICR - real variable containing value of Q_{iCR}

9. Printed Output: None

1. Program Name: SUBROUTINE RBMAX
2. Program Description: Subroutine RBMAX calculates the maximum depth of cut in feet that the propeller blade makes in a block of ice. This value is chosen at random from an array of values that represent the maximum depth of cut as a percentage of the propeller diameter.
3. Program Logic: The ice pattern is made up of two blocks of ice in a repeating sequence. Four variables control the spacing and size of the blocks (BLOC1, BLOC2, DBB12, and DBB21.) The value of the variable KELO determines where in the ice pattern the ship is at a given instance. The value of KELO is initially set to one (1) in the EXEC mainline. When this subroutine is initially executed, the ship had just entered block one and KELO is set to two (2). The variable XICE is used to save the ship's distance upon entering the ice to determine how long the ship will remain in a particular block of ice. The variable KELO assumes other values depending on whether the ship is between blocks one and two, in block two, or between blocks two and one. The first 10 values of the RBA array are used when in block one, the remaining 15 values when in block two.
4. Listing of Parameters: All parameters are in COMMON. Values taken from COMMON-KELO, CXS, BLOC1, RBA, D, DBB21, BLOC2, and DBB12. Results stored in COMMON - CSIE, SIE, RBMX, and KELO.
5. Program Length:
 - a. Source program - 87 statements (including comments).

b. Object program - 178 machine words.

6. Programs Called By This Program: None

7. Programs Calling This Program: Subroutine QICE

8. Local Variables:

XICE - real variable used to save the present value of the ship's distance each time a block is entered or exited from.

IA - integer variable used to reference the RBA array in determining the maximum depth of cut; updated by one (1) each time a new value is needed.

1. Program Name: SUBROUTINE STATE

2. Program Description: Subroutine STATE calculates the values of all the derivatives using the values calculated in the previous subroutines. The derivatives calculated are: 1) dz/dt -servo stroke velocity; 2) d^2z/dt^2 - servo stroke acceleration; 3) dn_{pm}/dt - time rate of prime mover speed; 4) di_{fg}/dt - time rate of generator field current; 4) di'_f/dt - time rate of motor field adjusting current; 6) dn_p/dt -propeller acceleration; 7) dv_s/dt - ship's acceleration; 8) dx_s/dt - ship's velocity; 9) SIEDO - peopeller angular velocity; and 10) di_a/dt - time rate of armature circuit current.

3. Program Logic: If the value of the actual bridge lever position (θ) is negative or if the bus voltage (e_b) is less than or equal to the constant C_1 , then the time rate of the motor field adjusting current is set to zero.

4. Listing of Parameters: All parameters are located in COMMON. Values taken from COMMON, Y, ALP1, ALPZ, ALP3, ALPD, XNEDO, XNE, Z, QDE, QG, PI, XJDG, EFG, RFG, RX, FGI, EG, EC, RA, AI, XLA, XGEN, XLASF, XM, XLFG, THETA, EB, C1, AIHAT, XLFM, QM, QFRM, QP, QI, XJMP, TP, R, XMASS, VS and XNP. Results stored in COMMON - ZDO, YDO, XNPMD, AIDO, FGIDO, FIPDO, XNPDO, VSDO, XSDO, and SIEDO.

5. Program Length:
 - a. Source program - 66 statements (including comments).
 - b. Object program - 252 machine words.

6. Programs Called By This Program: None

7. Program Calling This Program: Subroutine CALVR.

8. Local Variables:

VOLT1- real variable used to contain the value of
 $e_{fg} - (R_{fg} + R_x) i_{fg}$

VOLT2- real variable used to contain the value of
 $e_g - e_c - R_a i_a$

XLD - real variable used to contain the value of
 $L_a + 1.5 \cdot XLASF/XGEN$

XLM - real variable used to contain the value of
 $2.25 XM^2/XGEN$

DENOM - real variable used to contain the value of
 $1.5 L_{fg} (XLD-XLM)$

9. Printed Output: None

1. Program Name: SUBROUTINE OUT
2. Program Description: Subroutine OUT prints out the program variables in question on the IBM 1132 printer.
3. Program Logic: Before printing the variables in question, the following values are changed to different units: the value of the angular velocity of the prime mover (n_{pm}) is changed from revolutions per second to revolutions per minute as is the propeller angular velocity (n_p); the ship's speed (V_s) is converted from feet per second, to knots; the total propeller thrust (T_p) and ship resistance (R) are changed from pounds to long tons. The shaft horsepower is also computed from the motor torque (Q_M) and n_p .
4. Listing of Parameters: All parameters are in COMMON.
Values taken from COMMON - CXNPM, CXNP, CVS, TP, R, QM, PI, XNP, CT, CFGI, QDE, QG, EB, FMI, EC, AI, and QP.
Results stored in COMMON - None
5. Program Length:
 - a. Source program - 49 Statements (including comments).
 - b. Object program - 132 machine words.
6. Programs Called By This Program: None
7. Programs Calling This Program: EXEC mainline.
8. Local Variables:

RPMPM - real variable containing angular velocity of
 prime mover in revolutions per minute. '
 RPMP - real variable containing propeller angular
 velocity in revolutions per minute
 CVK - real variable containing ship's speed in knots.
 TPT - real variable containing total propeller thrust in
 long tons.
 RT - real variable containing ship's resistance in
 long tons.
 POW - Real variable containing ship's horsepower.

9. Printed Output: This program prints out the 16 variables in question on the IBM 1130 printer. The values occupy one line in the following format:

```

      /      RPMPM ifg      RPMP  CVK /  Qde
bxxx, xxxbxxx. x bxxx. xbxxxx. xbxxx. x xxxxxx. x

      Qg      db      ifm      ec      Qm
xxxxxx. xbxxx. xbxxx. xbxxx. bxxxxxxx. x

      ia      Qp      /  TPT  RT  /  POW
bxxxxxx. bxxxxxxx. x xxxxx. xbxxx. x xxxxxx. x
  
```


1. Program Name: SUBROUTINE FINIS
2. Program Description: Subroutine FINIS interrogates data switches 1 through 7 to achieve the following results:

Switch 1 on implies punch new initial condition cards, 2 on implies plot data, 3 on implies continue present solution, 4 on prevents future typing of data switch setting instructions. 5 on implies a new value of the time step (dt), 6 on implies a new value of the print interval, and 7 on implies a new value of the armature circuit inductance (L_a).
3. Program Logic: If a graph is indicated, the GRAPH mainline is loaded and executed by use of a CALL LINK (GRAPH) statement in this routine. The terminating time value (TEND) is set to the last time value considered before the graph routine is executed.
4. Listing of Parameters: All parameters are in COMMON.

Values taken from COMMON - THETA, XNPMD, Y, Z, FGI, XNPM, FIP, XNP, VS, AI, AIDQ, CT and DT.

Results stored in COMMON - TEND, DT, NON, XLA, and ISPAR(1)
5. Program Length:
 - a. Source program - 92 statements (including comments).
 - b. Object program - 498 machine words.
6. Programs Called By This Program: The GRAPH mainline may be loaded and executed by use of a CALL LINK (GRAPH) statement.
7. Programs Calling This Program: EXEC mainline

8. Local Variables: KEY1, KEY2, KEY3, KEY4, KEY5, KEY6, KEY7 - integer variables containing the switch settings for data switches 1 through 7 respectively. A value of one (1) indicates the switch was on, a value of 2 indicates off.
9. Printed Output: If switch 4 was off, the following switch setting instructions are typed out:

TURN ON SWITCH --

- 1 - TO PUNCH NEW INITIAL CONDITION CARDS
- 2 - TO PLOT DATA
- 3 - TO CONTINUE SOLUTION OF CURRENT PROBLEM -
SWITCH OFF STARTS NEW PROBLEM
- 4 - TO SUPPRESS FUTURE TYPING OF THESE 1
INSTRUCTIONS
- 5 - TO CHANGE TIME STEP
- 6 - TO CHANGE PRINT INTERVAL
- 7 - TO CHANGE XLA

TURN ON SELECTOR SWITCHES AND PRESS START

If new initial condition cards are punched, they are also printed on the IBM 1132 printer.

If new values of either dt , NON, or L_a are to be entered, any of the following messages, whichever be appropriate, will be typed out:

- () - TIME STEP
- () - NON
- () - XLA

1. Program Name: Mainline Routine GRAPH
2. Program Description: Mainline program GRAPH controls the plotting of the data stored on the disk by the EXEC mainline program. The program first requires that the code numbers of the variables to be plotted be entered from the console keyboard. A request is then made to determine if a grid is to be plotted by requiring a code number to be entered from the console keyboard. If a grid is to be plotted, the program requires three lines of graph identification to be entered from the keyboard after which the information as well as the graph serial number (computed or read in by the GENIS mainline) is printed out along with the grid. The program then loops through the array of code numbers to plot each variable in turn. If the curve extends beyond 20 seconds, the time axis plotter scale factor is reduced by a factor of 10 to plot the remainder of the curve. Upon the conclusion of the plotting for each curve, the pen is positioned for printing out the curve label. When all the requested variables have been plotted, the program loads and executes the GENIS mainline to begin a new problem.
3. Program Logic: When the curves extend beyond 20 seconds, it is necessary to compute a pseudo-origin for the remainder of the curve since the time axis plotter scale factor is reduced by a factor of 10 from the 20 second mark until the end of the graph. The new origin 'XDARN', is computed by multiplying the last time value encountered less than or equal to 19.9 seconds by 10 and subtracting the original value from the result. Following time values are added to this pseudo-origin to arrive at an augmented time value, 'CTT', for plotting the corresponding value of the variable in question on the curve.

4. Listing of Parameters: All parameters are located in COMMON. Values are taken from COMMON-SPARE(4), TEND, SX, SY, CT and VAR. Results stored in COMMON-NOVAR, NVAR, IELO, NIP, and JJ.
5. Program Length:
 - a. Source program - 246 statements (including comments).
 - b. Object program - 738 machine words.
6. Programs Called By This Program: Subroutines GRID, VAPLO, DATA, and PRINT. The GENIS mainline is loaded and executed by use of a CALL LINK (GENIS) statement following the completion of the entire graph.
7. Programs Calling This Program: This mainline is loaded and executed by use of the CALL LINK (GRAPH) statement in subroutine FINIS (of the EXEC mainline).
8. Local Variables:

NPLOT - this integer variable, although in COMMON, is local to this program and is used to contain the input from the keyboard that determines if a grid is to be plotted; a value greater than or equal to one (1) implies plot a grid, a value less than one (1) implies no grid is to be plotted.

TH - character rotation value: real variable containing the number of radians from horizontal (+X direction) that a line of characters is to be plotted.

L - integer variable used to control the up or down position of the pen during the plotting of a variable; initially set to one (1) so that the pen can be moved to the first value

to be plotted in the up-position for a particular variable, thereafter assumes a value of 2 since the pen will remain down in plotting the remainder of the curve.

XD, YD- real variables used to position the pen for plotting of the curve label after the curve has been plotted; set to start the first character 0.05 inches to the right and below the last value plotted.

The following local variables come into use only if the curves extend beyond 20 seconds:

SXX - time axis plotter scale factor: real variable set to one-tenth of the initial time axis plotter scale factor, 'SX', for plotting of the curves should they extend beyond 20 seconds since after 20 seconds the increments become 10 seconds.

XD - real variable set to 10 times the actual time value for the first time value exceeding or equal to 19.90 seconds.

XDARN- real variable set to the value of 'XD' minus the first time value exceeding or equal to 19.90 seconds. This value will then be used as a pseudo-origin for plotting the remainder of the curve since the plotter scale factor has been reduced by a factor of 10.

CTT - real variable equal to the actual time value, 'CT', plus the value of 'XDARN' for each time step in plotting a curve after the 20 second mark.

9. Printed Output: The following messages and requests for data are typed out on the console typewriter:

AI--1, EB--2, XNPM--3, FMI--4, XNP--5, XKW--6,
CVK--7, QPI--8, XPROI--9, CXS--10, QI--11, TEMP--12

WHICH VARIABLES TO BE PLOTTED (10 ONLY) -

() () () () () () () () ()

DO YOU WANT A GRID (0-NO, 1-YES)

()

If a grid is requested, three lines of not more than 50 characters
each are required to be input after which they will be plotted out
preceeded by the following information:

SERIAL NUMBER - - xxxxxxxxxxxx

1. Program Name: SUBROUTINE GRID
2. Program Description: Subroutine GRID is called to plot and label the time axis which is represented by the +X direction and the ordinate axis which is represented by the +Y and -Y directions. In plotting the time axis, the first 20 seconds are plotted in increments of one second. A test is then made to determine if the axis is to extend beyond 20 seconds and if so, the additional length is determined. The time axis is then plotted from the 20 second mark to the end in increments of 10 seconds. The ordinate axis is plotted from +5.0 to -5.0 with each unit representing one inch on the graph. The ordinate axis is labeled with the scale factors (quantity of each variable corresponding to one inch in the Y-direction) for the variables to be plotted.
3. Program Logic: In labeling the time axis, the local real variables 'XD' and 'WYE' are set to control the positioning of each label so that they will be 0.05 inches below and centered on each tick mark. If the time axis extends beyond 20 seconds, the additional length is computed as the number of tick marks beyond the one at 20 seconds (one less than the number of intervals between tick marks).
4. Listing of Parameters: All parameters are located in COMMON. Values taken from COMMON-SX, PI, TEND, NOVAR, NVAR, and SY. Results stored in COMMON-NIP.
5. Program Length:
 - a. Source program - 223 statements (including comments).
 - b. Object program - 696 machine words.
6. Programs Called By This Program: Subroutine VAPLO.

7. Programs Calling This Program: GRAPH mainline.

8. Local Variables:

- SXX - time axis plotter scale factor: real variable initially set to 'SX' for the plotting of the first 20 seconds of the time axis and then to one-tenth of 'SX' for the plotting of the remainder of the time axis since after 20 seconds the time interval becomes 10 seconds over the same actual distance initially used for the one (1) second intervals.
- TH - character rotation value: real variable containing the number of radians from horizontal (+X direction) that a line of characters is to be plotted.
- XD - label centering value: real variable used to center the axes labels at each tick mark; equal to 0.05 divided by the plotter scale factor for the axis in question.
- WYE - real variable used initially to place the X-axis labels 0.05 inches below each tick mark and then to indicate the relative position of each label for the +Y and -Y directions.
- X - real variable initially used to label the time axis and then used to start the Y-axis labels 0.77 inches to the left of the Y-axis. Finally used to indicate the starting position (to the left of the Y-axis) that each curve actual scale factor will be listed on the graph.
- TEST - real variable equal to the extent of the time axis beyond 20 seconds.
- TEST1 - real variable equal to one tenth of the extent of the time axis beyond 20 seconds plus a constant of one (1) to plot out the remaining tick marks.

ITEST - integer variable containing the number of tick marks to be plotted beyond the one at the 20 second mark.

I - integer variable used as a DO-loop index in labeling the axes.

MIKE - integer variable used to store in turn the code number for each variable to be plotted for use in a computed GO TO statement to plot the actual scale factor for that variable.

SY1 - real value containing the actual scale factor (quantity of variable per inch along the Y-axis) for each variable in turn as the scale factor is plotted.

9. Printed Output: Any of the following 12 statements concerning the Y-axis scale factors may be listed on the graph:

AI (1 IN. = xxxxx. x AMPS)

EB (1 IN. = xxx. x VOLTS)

NPM (1 IN. = xxx. x RPM)

FMI (1 IN. = xxx. x AMPS)

NP (1 IN. = xxx. x RPM)

KW (1 IN. = xxxx. x KW)

VK (1 IN. = xx. x K'TS.)

QP (1 IN. = xxxxxx. x FT-LBS)

TP (1 IN. = xxxxxx. x LBS)

XS (1 IN. = xxxx. x FEET)

QI (1 IN. = xxxxxx. x FT-LBS)

TEMP (1 IN. = xxx. x DEG. CENT.)

The following time axis label is plotted out:

TIME (SECS)

Each time axis tick mark is labeled starting at 1.0 and going through 20.0 by one's and then through the final value by 10's. Every whole ordinate (Y) axis tick mark is labeled from 5.0 through 0.0 to -5.0.

1. Program Name: SUBROUTINE DATA
2. Program Description: Subroutine DATA retrieves from the disk file the next time value and next value of the variable currently being plotted. The disk file index, 'IELO', is automatically updated to the next record after the values are retrieved. The time value is placed in location 'CT' and the new value of the variable being plotted in 'VAR', both in COMMON.
3. Program Logic: The current value of the disk file index ('IELO') is used to reference the record of the disk file that is next to be retrieved. Since there is no way to read just one value from the disk, the entire record must be retrieved, after which the value of 'IELO' is automatically updated by one (1) to point to the next record. The code number of the variable currently being plotted is found in the word of the variable list array, 'NVAR', currently being considered which is indicated by value of the variable list array index 'JJ'. A computed GO TO statement is then used to return the new value for the variable in question.
4. Listing of Parameters: All parameters are located in COMMON. Values taken from COMMON-NVAR, JJ, and IELO. Results stored in COMMON -IELO, CT, and VAR.
5. Program Length:
 - a. Source program - 90 statements (including comments).
 - b. Object program - 136 machine words.
6. Programs Called By This Program: None
7. Programs Calling This Program: GRAPH mainline

8. Local Variables:

NOOK - integer variable used to store the code number of the variable in question for use in a computed GO TO statement.

The following real variables, although appearing in COMMON, are actually local to this program and are used to store the value of each variable when the file record is read from the disk:

AI -	armature circuit current
EB-	bus volts
RPMPM-	diesel speed
FMI -	motor field current
RPMP-	propeller speed
XKW -	power
CVK -	vessel speed
QPI -	propeller torque
XPROI -	propeller thrust
CXS -	distance
QI -	ice torque
TEMP-	motor temperature

9. Printed Output: None

1. Program Name: SUBROUTINE VAPLO
2. Program Description: Subroutine VAPLO retrieves the plotter scale factor for the variable that is about to be plotted. The plotter scale factor is the actual number of inches along the Y-axis that corresponds to one unit of the variable in question and is placed in 'SY'.
3. Program Logic: The code number of the variable about to be plotted is found in the word of the variable list array, 'NVAR', that is currently being examined which is indicated by the value of the variable list array index, 'NIP'. A computed GO TO statement is then used to return the appropriate plotter scale factor.
4. Listing of Parameters: All parameters are located in COMMON. Values taken from COMMON-NVAR, NIP, SY1, SY2, SY3, SY4, SY5, SY6, SY7, SY8, SY9, SY10, SY11, and SY12. Results stored in COMMON - SY.
5. Program Length:
 - a. Source program - 85 statements (including comments).
 - b. Object program - 100 machine words.
6. Programs Called By This Program: None
7. Programs Calling This Program: GRAPH mainline, subroutine GRID.
8. Local Variables:

NAN - integer variable used to store the code number of the variable in question for use in a computed GO TO statement.
9. Printed Output: None

1. Program Name: SUBROUTINE PRINT
2. Program Description: Subroutine PRINT is executed upon the completion of the curve for a particular variable in order to plot the appropriate curve label for that variable. The curve label is plotted starting at the current location of the pen which was set by the GRAPH mainline before calling this routine to be immediately following the last value plotted for the variable.
3. Program Logic: The code number of the variable in question is found in the word of the variable list array, 'NVAR', that is currently being examined which is indicated by the value of the variable list array index 'JJ'. A computed GO TO statement is then used to plot out the appropriate curve label.
4. Listing of Parameters: All parameters are located in COMMON. Values taken from COMMON - NVAR and JJ. Results stored in COMMON - none.
5. Program Length:
 - a. Source program - 97 statements (including comments).
 - b. Object program - 212 machine words.
6. Programs Called By This Program: None
7. Programs Calling This Program: GRAPH mainline
8. Local Variables:

MIKE - integer variable used to store the code number of the variable in question for use in a computed GO TO statement.

9. Printed Output: This program may plot out any one of the following 12 curve labels depending on the variable in question:

- ARM. CIR. CUR.
- BUS VOLTS
- DIESEL SPEED
- MOTOR FIELD CUR.
- PROP SPEED
- POWER
- VESSEL SPEED
- PROP TORQUE
- PROP THRUST
- DISTANCE
- ICE TORQUE
- MOTOR TEMP.

PROGRAM LISTINGS


```

// FOR GENIS
// HAVE WORD INTEGERS
// LIST ALL
// CLOS(CARD,TYPE,WRITE,KEYBOARD,1132PRINTER,DISK)
// GENIS MAINLINE
// VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968
//
// DIMENSION LIST(140)
// COMMON RO,PI,IZ,IZBAR,DDR,D,XJPA,XJY,XJSH,XJD,XJG,EFG,
// 1 XJVR,XJOG,FIWAX,ZA,PG,CI,XK1,OBGR,CSHER,CW,PCVS,XLA,
// 2 TAU1,TAU2,XKS,ALP1,ALP2,ALP3,ALP4,XVASS
// COMMON NOVAR,NVAR(10),JU,ICNTR,NPLOT,NCATA,NOP,NON,NIP
// COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12
// COMMON OT,CT,T,TEND,TIVSC
// COMMON Z,V,XNRM,FGI,FIP,XNP,VS,XS,SIE,AI,TV,TEMP
// COMMON CZ,CY,CXNEN,CEGL,CEIP,CXNP,CVS,CXS,CS:E,CAL,CRV,CTEMP
// COMMON ZDO,YDS,XNPD,FGIDO,FIPO,XNPD,VSD,XSD,SIEDO,AIDO,RVDO,
// 1 TVDO
// COMMON XNE,XNEDO,VA,THETO,THET1,THETA,THETO,THETD,GG,EG,ER,EC
// 10X,DERM,ALPAT,CP,CP1,CP2,CP3,CP4,CP5,CP6,CP7,CP8,CP9,CP10,CP11,CP12,CP13,CP14,CP15,CP16,CP17,CP18,CP19,CP20,CP21,CP22,CP23,CP24,CP25,CP26,CP27,CP28,CP29,CP30,CP31,CP32,CP33,CP34,CP35,CP36,CP37,CP38,CP39,CP40,CP41,CP42,CP43,CP44,CP45,CP46,CP47,CP48,CP49,CP50,CP51,CP52,CP53,CP54,CP55,CP56,CP57,CP58,CP59,CP60,CP61,CP62,CP63,CP64,CP65,CP66,CP67,CP68,CP69,CP70,CP71,CP72,CP73,CP74,CP75,CP76,CP77,CP78,CP79,CP80,CP81,CP82,CP83,CP84,CP85,CP86,CP87,CP88,CP89,CP90,CP91,CP92,CP93,CP94,CP95,CP96,CP97,CP98,CP99,CP100,CP101,CP102,CP103,CP104,CP105,CP106,CP107,CP108,CP109,CP110,CP111,CP112,CP113,CP114,CP115,CP116,CP117,CP118,CP119,CP120,CP121,CP122,CP123,CP124,CP125,CP126,CP127,CP128,CP129,CP130,CP131,CP132,CP133,CP134,CP135,CP136,CP137,CP138,CP139,CP140,CP141,CP142,CP143,CP144,CP145,CP146,CP147,CP148,CP149,CP150,CP151,CP152,CP153,CP154,CP155,CP156,CP157,CP158,CP159,CP160,CP161,CP162,CP163,CP164,CP165,CP166,CP167,CP168,CP169,CP170,CP171,CP172,CP173,CP174,CP175,CP176,CP177,CP178,CP179,CP180,CP181,CP182,CP183,CP184,CP185,CP186,CP187,CP188,CP189,CP190,CP191,CP192,CP193,CP194,CP195,CP196,CP197,CP198,CP199,CP200,CP201,CP202,CP203,CP204,CP205,CP206,CP207,CP208,CP209,CP210,CP211,CP212,CP213,CP214,CP215,CP216,CP217,CP218,CP219,CP220,CP221,CP222,CP223,CP224,CP225,CP226,CP227,CP228,CP229,CP230,CP231,CP232,CP233,CP234,CP235,CP236,CP237,CP238,CP239,CP240,CP241,CP242,CP243,CP244,CP245,CP246,CP247,CP248,CP249,CP250,CP251,CP252,CP253,CP254,CP255,CP256,CP257,CP258,CP259,CP260,CP261,CP262,CP263,CP264,CP265,CP266,CP267,CP268,CP269,CP270,CP271,CP272,CP273,CP274,CP275,CP276,CP277,CP278,CP279,CP280,CP281,CP282,CP283,CP284,CP285,CP286,CP287,CP288,CP289,CP290,CP291,CP292,CP293,CP294,CP295,CP296,CP297,CP298,CP299,CP300,CP301,CP302,CP303,CP304,CP305,CP306,CP307,CP308,CP309,CP310,CP311,CP312,CP313,CP314,CP315,CP316,CP317,CP318,CP319,CP320,CP321,CP322,CP323,CP324,CP325,CP326,CP327,CP328,CP329,CP330,CP331,CP332,CP333,CP334,CP335,CP336,CP337,CP338,CP339,CP340,CP341,CP342,CP343,CP344,CP345,CP346,CP347,CP348,CP349,CP350,CP351,CP352,CP353,CP354,CP355,CP356,CP357,CP358,CP359,CP360,CP361,CP362,CP363,CP364,CP365,CP366,CP367,CP368,CP369,CP370,CP371,CP372,CP373,CP374,CP375,CP376,CP377,CP378,CP379,CP380,CP381,CP382,CP383,CP384,CP385,CP386,CP387,CP388,CP389,CP390,CP391,CP392,CP393,CP394,CP395,CP396,CP397,CP398,CP399,CP400,CP401,CP402,CP403,CP404,CP405,CP406,CP407,CP408,CP409,CP410,CP411,CP412,CP413,CP414,CP415,CP416,CP417,CP418,CP419,CP420,CP421,CP422,CP423,CP424,CP425,CP426,CP427,CP428,CP429,CP430,CP431,CP432,CP433,CP434,CP435,CP436,CP437,CP438,CP439,CP440,CP441,CP442,CP443,CP444,CP445,CP446,CP447,CP448,CP449,CP450,CP451,CP452,CP453,CP454,CP455,CP456,CP457,CP458,CP459,CP460,CP461,CP462,CP463,CP464,CP465,CP466,CP467,CP468,CP469,CP470,CP471,CP472,CP473,CP474,CP475,CP476,CP477,CP478,CP479,CP480,CP481,CP482,CP483,CP484,CP485,CP486,CP487,CP488,CP489,CP490,CP491,CP492,CP493,CP494,CP495,CP496,CP497,CP498,CP499,CP500,CP501,CP502,CP503,CP504,CP505,CP506,CP507,CP508,CP509,CP510,CP511,CP512,CP513,CP514,CP515,CP516,CP517,CP518,CP519,CP520,CP521,CP522,CP523,CP524,CP525,CP526,CP527,CP528,CP529,CP530,CP531,CP532,CP533,CP534,CP535,CP536,CP537,CP538,CP539,CP540,CP541,CP542,CP543,CP544,CP545,CP546,CP547,CP548,CP549,CP550,CP551,CP552,CP553,CP554,CP555,CP556,CP557,CP558,CP559,CP560,CP561,CP562,CP563,CP564,CP565,CP566,CP567,CP568,CP569,CP570,CP571,CP572,CP573,CP574,CP575,CP576,CP577,CP578,CP579,CP580,CP581,CP582,CP583,CP584,CP585,CP586,CP587,CP588,CP589,CP590,CP591,CP592,CP593,CP594,CP595,CP596,CP597,CP598,CP599,CP600,CP601,CP602,CP603,CP604,CP605,CP606,CP607,CP608,CP609,CP610,CP611,CP612,CP613,CP614,CP615,CP616,CP617,CP618,CP619,CP620,CP621,CP622,CP623,CP624,CP625,CP626,CP627,CP628,CP629,CP630,CP631,CP632,CP633,CP634,CP635,CP636,CP637,CP638,CP639,CP640,CP641,CP642,CP643,CP644,CP645,CP646,CP647,CP648,CP649,CP650,CP651,CP652,CP653,CP654,CP655,CP656,CP657,CP658,CP659,CP660,CP661,CP662,CP663,CP664,CP665,CP666,CP667,CP668,CP669,CP670,CP671,CP672,CP673,CP674,CP675,CP676,CP677,CP678,CP679,CP680,CP681,CP682,CP683,CP684,CP685,CP686,CP687,CP688,CP689,CP690,CP691,CP692,CP693,CP694,CP695,CP696,CP697,CP698,CP699,CP700,CP701,CP702,CP703,CP704,CP705,CP706,CP707,CP708,CP709,CP710,CP711,CP712,CP713,CP714,CP715,CP716,CP717,CP718,CP719,CP720,CP721,CP722,CP723,CP724,CP725,CP726,CP727,CP728,CP729,CP730,CP731,CP732,CP733,CP734,CP735,CP736,CP737,CP738,CP739,CP740,CP741,CP742,CP743,CP744,CP745,CP746,CP747,CP748,CP749,CP750,CP751,CP752,CP753,CP754,CP755,CP756,CP757,CP758,CP759,CP760,CP761,CP762,CP763,CP764,CP765,CP766,CP767,CP768,CP769,CP770,CP771,CP772,CP773,CP774,CP775,CP776,CP777,CP778,CP779,CP780,CP781,CP782,CP783,CP784,CP785,CP786,CP787,CP788,CP789,CP790,CP791,CP792,CP793,CP794,CP795,CP796,CP797,CP798,CP799,CP800,CP801,CP802,CP803,CP804,CP805,CP806,CP807,CP808,CP809,CP810,CP811,CP812,CP813,CP814,CP815,CP816,CP817,CP818,CP819,CP820,CP821,CP822,CP823,CP824,CP825,CP826,CP827,CP828,CP829,CP830,CP831,CP832,CP833,CP834,CP835,CP836,CP837,CP838,CP839,CP840,CP841,CP842,CP843,CP844,CP845,CP846,CP847,CP848,CP849,CP850,CP851,CP852,CP853,CP854,CP855,CP856,CP857,CP858,CP859,CP860,CP861,CP862,CP863,CP864,CP865,CP866,CP867,CP868,CP869,CP870,CP871,CP872,CP873,CP874,CP875,CP876,CP877,CP878,CP879,CP880,CP881,CP882,CP883,CP884,CP885,CP886,CP887,CP888,CP889,CP890,CP891,CP892,CP893,CP894,CP895,CP896,CP897,CP898,CP899,CP900,CP901,CP902,CP903,CP904,CP905,CP906,CP907,CP908,CP909,CP910,CP911,CP912,CP913,CP914,CP915,CP916,CP917,CP918,CP919,CP920,CP921,CP922,CP923,CP924,CP925,CP926,CP927,CP928,CP929,CP930,CP931,CP932,CP933,CP934,CP935,CP936,CP937,CP938,CP939,CP940,CP941,CP942,CP943,CP944,CP945,CP946,CP947,CP948,CP949,CP950,CP951,CP952,CP953,CP954,CP955,CP956,CP957,CP958,CP959,CP960,CP961,CP962,CP963,CP964,CP965,CP966,CP967,CP968,CP969,CP970,CP971,CP972,CP973,CP974,CP975,CP976,CP977,CP978,CP979,CP980,CP981,CP982,CP983,CP984,CP985,CP986,CP987,CP988,CP989,CP990,CP991,CP992,CP993,CP994,CP995,CP996,CP997,CP998,CP999,CP1000,CP1001,CP1002,CP1003,CP1004,CP1005,CP1006,CP1007,CP1008,CP1009,CP1010,CP1011,CP1012,CP1013,CP1014,CP1015,CP1016,CP1017,CP1018,CP1019,CP1020,CP1021,CP1022,CP1023,CP1024,CP1025,CP1026,CP1027,CP1028,CP1029,CP1030,CP1031,CP1032,CP1033,CP1034,CP1035,CP1036,CP1037,CP1038,CP1039,CP1040,CP1041,CP1042,CP1043,CP1044,CP1045,CP1046,CP1047,CP1048,CP1049,CP1050,CP1051,CP1052,CP1053,CP1054,CP1055,CP1056,CP1057,CP1058,CP1059,CP1060,CP1061,CP1062,CP1063,CP1064,CP1065,CP1066,CP1067,CP1068,CP1069,CP1070,CP1071,CP1072,CP1073,CP1074,CP1075,CP1076,CP1077,CP1078,CP1079,CP1080,CP1081,CP1082,CP1083,CP1084,CP1085,CP1086,CP1087,CP1088,CP1089,CP1090,CP1091,CP1092,CP1093,CP1094,CP1095,CP1096,CP1097,CP1098,CP1099,CP1100,CP1101,CP1102,CP1103,CP1104,CP1105,CP1106,CP1107,CP1108,CP1109,CP1110,CP1111,CP1112,CP1113,CP1114,CP1115,CP1116,CP1117,CP1118,CP1119,CP1120,CP1121,CP1122,CP1123,CP1124,CP1125,CP1126,CP1127,CP1128,CP1129,CP1130,CP1131,CP1132,CP1133,CP1134,CP1135,CP1136,CP1137,CP1138,CP1139,CP1140,CP1141,CP1142,CP1143,CP1144,CP1145,CP1146,CP1147,CP1148,CP1149,CP1150,CP1151,CP1152,CP1153,CP1154,CP1155,CP1156,CP1157,CP1158,CP1159,CP1160,CP1161,CP1162,CP1163,CP1164,CP1165,CP1166,CP1167,CP1168,CP1169,CP1170,CP1171,CP1172,CP1173,CP1174,CP1175,CP1176,CP1177,CP1178,CP1179,CP1180,CP1181,CP1182,CP1183,CP1184,CP1185,CP1186,CP1187,CP1188,CP1189,CP1190,CP1191,CP1192,CP1193,CP1194,CP1195,CP1196,CP1197,CP1198,CP1199,CP1200,CP1201,CP1202,CP1203,CP1204,CP1205,CP1206,CP1207,CP1208,CP1209,CP1210,CP1211,CP1212,CP1213,CP1214,CP1215,CP1216,CP1217,CP1218,CP1219,CP1220,CP1221,CP1222,CP1223,CP1224,CP1225,CP1226,CP1227,CP1228,CP1229,CP1230,CP1231,CP1232,CP1233,CP1234,CP1235,CP1236,CP1237,CP1238,CP1239,CP1240,CP1241,CP1242,CP1243,CP1244,CP1245,CP1246,CP1247,CP1248,CP1249,CP1250,CP1251,CP1252,CP1253,CP1254,CP1255,CP1256,CP1257,CP1258,CP1259,CP1260,CP1261,CP1262,CP1263,CP1264,CP1265,CP1266,CP1267,CP1268,CP1269,CP1270,CP1271,CP1272,CP1273,CP1274,CP1275,CP1276,CP1277,CP1278,CP1279,CP1280,CP1281,CP1282,CP1283,CP1284,CP1285,CP1286,CP1287,CP1288,CP1289,CP1290,CP1291,CP1292,CP1293,CP1294,CP1295,CP1296,CP1297,CP1298,CP1299,CP1300,CP1301,CP1302,CP1303,CP1304,CP1305,CP1306,CP1307,CP1308,CP1309,CP1310,CP1311,CP1312,CP1313,CP1314,CP1315,CP1316,CP1317,CP1318,CP1319,CP1320,CP1321,CP1322,CP1323,CP1324,CP1325,CP1326,CP1327,CP1328,CP1329,CP1330,CP1331,CP1332,CP1333,CP1334,CP1335,CP1336,CP1337,CP1338,CP1339,CP1340,CP1341,CP1342,CP1343,CP1344,CP1345,CP1346,CP1347,CP1348,CP1349,CP1350,CP1351,CP1352,CP1353,CP1354,CP1355,CP1356,CP1357,CP1358,CP1359,CP1360,CP1361,CP1362,CP1363,CP1364,CP1365,CP1366,CP1367,CP1368,CP1369,CP1370,CP1371,CP1372,CP1373,CP1374,CP1375,CP1376,CP1377,CP1378,CP1379,CP1380,CP1381,CP1382,CP1383,CP1384,CP1385,CP1386,CP1387,CP1388,CP1389,CP1390,CP1391,CP1392,CP1393,CP1394,CP1395,CP1396,CP1397,CP1398,CP1399,CP1400,CP1401,CP1402,CP1403,CP1404,CP1405,CP1406,CP1407,CP1408,CP1409,CP1410,CP1411,CP1412,CP1413,CP1414,CP1415,CP1416,CP1417,CP1418,CP1419,CP1420,CP1421,CP1422,CP1423,CP1424,CP1425,CP1426,CP1427,CP1428,CP1429,CP1430,CP1431,CP1432,CP1433,CP1434,CP1435,CP1436,CP1437,CP1438,CP1439,CP1440,CP1441,CP1442,CP1443,CP1444,CP1445,CP1446,CP1447,CP1448,CP1449,CP1450,CP1451,CP1452,CP1453,CP1454,CP1455,CP1456,CP1457,CP1458,CP1459,CP1460,CP1461,CP1462,CP1463,CP1464,CP1465,CP1466,CP1467,CP1468,CP1469,CP1470,CP1471,CP1472,CP1473,CP1474,CP1475,CP1476,CP1477,CP1478,CP1479,CP1480,CP1481,CP1482,CP1483,CP1484,CP1485,CP1486,CP1487,CP1488,CP1489,CP1490,CP1491,CP1492,CP1493,CP1494,CP1495,CP1496,CP1497,CP1498,CP1499,CP1500,CP1501,CP1502,CP1503,CP1504,CP1505,CP1506,CP1507,CP1508,CP1509,CP1510,CP1511,CP1512,CP1513,CP1514,CP1515,CP1516,CP1517,CP1518,CP1519,CP1520,CP1521,CP1522,CP1523,CP1524,CP1525,CP1526,CP1527,CP1528,CP1529,CP1530,CP1531,CP1532,CP1533,CP1534,CP1535,CP1536,CP1537,CP1538,CP1539,CP1540,CP1541,CP1542,CP1543,CP1544,CP1545,CP1546,CP1547,CP1548,CP1549,CP1550,CP1551,CP1552,CP1553,CP1554,CP1555,CP1556,CP1557,CP1558,CP1559,CP1560,CP1561,CP1562,CP1563,CP1564,CP1565,CP1566,CP1567,CP1568,CP1569,CP1570,CP1571,CP1572,CP1573,CP1574,CP1575,CP1576,CP1577,CP1578,CP1579,CP1580,CP1581,CP1582,CP1583,CP1584,CP1585,CP1586,CP1587,CP1588,CP1589,CP1590,CP1591,CP1592,CP1593,CP1594,CP1595,CP1596,CP1597,CP1598,CP1599,CP1600,CP1601,CP1602,CP1603,CP1604,CP1605,CP1606,CP1607,CP1608,CP1609,CP1610,CP1611,CP1612,CP1613,CP1614,CP1615,CP1616,CP1617,CP1618,CP1619,CP1620,CP1621,CP1622,CP1623,CP1624,CP1625,CP1626,CP1627,CP1628,CP1629,CP1630,CP1631,CP1632,CP1633,CP1634,CP1635,CP1636,CP1637,CP1638,CP1639,CP1640,CP1641,CP1642,CP1643,CP1644,CP1645,CP1646,CP1647,CP1648,CP1649,CP1650,CP1651,CP1652,CP1653,CP1654,CP1655,CP1656,CP1657,CP1658,CP1659,CP1660,CP1661,CP1662,CP1663,CP1664,CP1665,CP1666,CP1667,CP1668,CP1669,CP1670,CP1671,CP1672,CP1673,CP1674,CP1675,CP1676,CP1677,CP1678,CP1679,CP1680,CP1681,CP1682,CP1683,CP1684,CP1685,CP1686,CP1687,CP1688,CP1689,CP1690,CP1691,CP1692,CP1693,CP1694,CP1695,CP1696,CP1697,CP1698,CP1699,CP1700,CP1701,CP1702,CP1703,CP1704,CP1705,CP1706,CP1707,CP1708,CP1709,CP1710,CP1711,CP1712,CP1713,CP1714,CP1715,CP1716,CP1717,CP1718,CP1719,CP1720,CP1721,CP1722,CP1723,CP1724,CP1725,CP1726,CP1727,CP1728,CP1729,CP1730,CP1731,CP1732,CP1733,CP1734,CP1735,CP1736,CP1737,CP1738,CP1739,CP1740,CP1741,CP1742,CP1743,CP1744,CP1745,CP1746,CP1747,CP1748,CP1749,CP1750,CP1751,CP1752,CP1753,CP1754,CP1755,CP1756,CP1757,CP1758,CP1759,CP1760,CP1761,CP1762,CP1763,CP1764,CP1765,CP1766,CP1767,CP1768,CP1769,CP1770,CP1771,CP1772,CP1773,CP1774,CP1775,CP1776,CP1777,CP1778,CP1779,CP1780,CP1781,CP1782,CP1783,CP1784,CP1785,CP1786,CP1787,CP1788,CP1789,CP1790,CP1791,CP1792,CP1793,CP1794,CP1795,CP1796,CP1797,CP1798,CP1799,CP1800,CP1801,CP1802,CP1803,CP1804,CP1805,CP1806,CP1807,CP1808,CP1809,CP1810,CP1811,CP1812,CP1813,CP1814,CP1815,CP1816,CP1817,CP1818,CP1819,CP1820,CP1821,CP1822,CP1823,CP1824,CP1825,CP1826,CP1827,CP1828,CP1829,CP1830,CP1831,CP1832,CP1833,CP1834,CP1835,CP1836,CP1837,CP1838,CP1839,CP1840,CP1841,CP1842,CP1843,CP1844,CP1845,CP1846,CP1847,CP1848,CP1849,CP1850,CP1851,CP1852,CP1853,CP1854,CP1855,CP1856,CP1857,CP1858,CP1859,CP1860,CP1861,CP1862,CP1863,CP1864,CP1865,CP1866,CP1867,CP1868,CP1869,CP1870,CP1871,CP1872,CP1873,CP1874,CP1875,CP1876,CP1877,CP1878,CP1879,CP1880,CP1881,CP1882,CP1883,CP1884,CP1885,CP1886,CP1887,CP1888,CP1889,CP1890,CP1891,CP1892,CP1893,CP1894,CP1895,CP1896,CP1897,CP1898,CP1899,CP1900,CP1901,CP1902,CP1903,CP1904,CP1905,CP1906,CP1907,CP1908,CP1909,CP1910,CP1911,CP1912,CP1913,CP1914,CP1915,CP1916,CP1917,CP1918,CP1919,CP1920,CP1921,CP1922,CP1923,CP1924,CP1925,CP1926,CP1927,CP1928,CP1929,CP1930,CP1931,CP1932,CP1933,CP1934,CP1935,CP1936,CP1937,CP1938,CP1939,CP1940,CP1941,CP1942,CP1943,CP1944,CP1945,CP1946,CP1947,CP1948,CP1949,CP1950,CP1951,CP1952,CP1953,CP1954,CP1955,CP1956,CP1957,CP1958,CP1959,CP1960,CP1961,CP1962,CP1963,CP1964,CP1965,CP1966,CP1967,CP1968,CP1969,CP1970,CP1971,CP1972,CP1973,CP1974,CP1975,CP1976,CP1977,CP1978,CP1979,CP1980,CP1981,CP1982,CP1983,CP1984,CP1985,CP1986,CP1987,CP1988,CP1989,CP1990,CP1991,CP1992,CP1993,CP1994,CP1995,CP1996,CP1997,CP1998,CP1999,CP2000,CP2001,CP2002,CP2003,CP2004,CP2005,CP2006,CP2007,CP2008,CP2009,CP2010,CP2011,CP2012,CP2013,CP2014,CP2015,CP2016,CP2017,CP2018,CP2019,CP2020,CP2021,CP2022,CP2023,CP2024,CP2025,CP2026,CP2027,CP2028,CP2029,CP2030,CP2031,CP2032,CP2033,CP2034,CP2035,CP2036,CP2037,CP2038,CP2039,CP2040,CP2041,CP2042,CP2043,CP2044,CP2045,CP2046,CP2047,CP2048,CP2049,CP2050,CP2051,CP2052,CP2053,CP2054,CP2055,CP2056,CP2057,CP2058,CP2059,CP2060,CP2061,CP2062,CP2063,CP2064,CP2065,CP2066,CP2067,CP2068,CP2069,CP2070,CP2071,CP2072,CP2073,CP2074,CP2075,CP2076,CP2077,CP2078,CP2079,CP2080,CP2081,CP2082,CP2083,CP2084,CP2085,CP2086,CP2087,CP2088,CP2089,CP2090,CP2091,CP2092,CP2093,CP2094,CP2095,CP2096,CP2097,CP2098,CP2099,CP2100,CP2101,CP2102,CP2103,CP2104,CP2105,CP2106,CP2107,CP2108,CP2109,CP2110,CP2111,CP2112,CP2113,CP2114,CP2115,CP2116,CP2117,CP2118,CP2119,CP2120,CP2121,CP2122,CP2123,CP2124,CP2125,CP2126,CP2127,CP2128,CP2129,CP2130,CP2131,CP2132,CP2133,CP2134,CP2135,CP2136,CP2137,CP2138,CP2139,CP2140,CP2141,CP2142,CP2143,CP2144,CP2145,CP2146,CP2147,CP2148,CP2149,CP2150,CP2151,CP2152,CP2153,CP2154,CP2155,CP2156,CP2157,CP2158,CP2159,CP2160,CP2161,CP2162,CP2163,CP2164,CP2165,CP2166,CP2167,CP2168,CP2169,CP2170,CP2171,CP2172,CP2173,CP2174,CP2175
```



```

1 ACTERS OR LESS')
2 READ(6,9) (LIST(I),I=1,40)
3 FORMAT(40A2)
4 WRITE(3,4) (LIST(I),I=1,40)
5 FORMAT(//////////, 'SIMULATION OF ',40A2)
6 CHECK IF SERIAL NUMBER IS TO BE UPDATED.
7 RESET TO ZERO IF DATA SWITCH 9 ON
8
9 CALL DATSW(9,KEY9)
10 GO TO(80,81),KEY9
11 SER=0.0
12 WRITE(2,1) SER
13 READ(2,1) SER
14 SER=SER+1.
15 WRITE(3,82) SER
16 FORMAT(' SERIAL NUMBER--',F10.0)
17 WRITE(2,1) SER
18 WRITE(3,9)
19 FORMAT(14I,5X,'SELECTED INPUT DATA')
20
21 ***** PRINT OUT INITIAL CONDITIONS
22
23 WRITE(3,5)
24 FORMAT(//////,1X,'INITIAL CONDITIONS',/)
25 WRITE(3,100)
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```



```

CALL DATSW (4,KEY4)
GO TO (1191,1151), KEY4
1151 WRITE(1,15)
15 FORMAT('TURN ON APPROPRIATE SWITCHES')
116 WRITE(1,115) ICE DATA- 10 ' )
16 WRITE(1,16) PROP DATA- 11 ' )
17 WRITE(1,17) HULL DATA- 12 ' )
18 WRITE(1,18) ELECTRICAL DATA- 13 ' )
19 WRITE(1,19) GOVERNOR AND BRIDGE CONTROLLER DATA- 14 ' )
119 WRITE(1,119) INERTIAS- 15 ' )
1191 WRITE(1,1192)
1192 FORMAT('/////',' TURN ON SELECTOR SWITCHES AND PRESS START')
PAUSE
CALL DATSW(11,1)
IF(1-1) 30,20,30
C*****
C*****
C*****
20 WRITE(3,21)
21 FORMAT('///,1X,'PROP DATA')
22 WRITE(3,22) 12B,DAR,F0,DCT
22 FORMAT('1X,'179='15,'
DAR='F10.4,' EO='F10.4,'
1DCT='F10.4)
23 WRITE(3,23) PDR,DSEK
23 FORMAT(1X,'PDR='F10.4,' D ='F10.4,' EK='F10.4)
24 WRITE(3,24)
24 FORMAT(9X,'R',20X,'SR',18X,'E',19X,'EE')
DO 26 I=1,32
WRITE(3,25) 9(1),99(1),E(1),EE(1)
25 FORMAT(1X,F10.4,11X,F10.4,11X,F10.4,11X,F10.4)
26 CONTINUE
DO 28 I=33,64
WRITE(3,27) 9(1),99(1)
27 FORMAT(1X,F10.4,11X,F10.4)
28 CONTINUE
30 CALL TSW(12,J)
IF(J-1) 40,31,40
C*****
C*****
C*****
31 WRITE(3,32)
32 FORMAT('///,1X,'HULL DATA')
33 WRITE(3,33) CAD,DELTA,XXX
33 FORMAT('1X,'CAD='F10.4,' DELTA='F10.1,' XXX='F10.0)
34 WRITE(3,34)
34 FORMAT(2X,'VK',15X,'EMP',20X,'WADED',20X,'THDED')
DO 37 I=1,20
K=I-1

```

GEN11080
 GEN11090
 GEN11100
 GEN11110
 GEN11120
 GEN11130
 GEN11140
 GEN11150
 GEN11160
 GEN11170
 GEN11180
 GEN11190
 GEN11200
 GEN11210
 GEN11220
 GEN11230
 GEN11240
 GEN11250
 GEN11260
 GEN11270
 GEN11280
 GEN11290
 GEN11300
 GEN11310
 GEN11320
 GEN11330
 GEN11340
 GEN11350
 GEN11360
 GEN11370
 GEN11380
 GEN11390
 GEN11400
 GEN11410
 GEN11420
 GEN11430
 GEN11440
 GEN11450
 GEN11460
 GEN11470
 GEN11480
 GEN11490
 GEN11500
 GEN11510
 GEN11520
 GEN11530
 GEN11540
 GEN11550
 GEN11560
 GEN11570
 GEN11580
 GEN11590
 GEN11600
 GEN11610


```

WRITE(3,36) X,EMP(1),WADED(1),THDED(1)
36 FORMAT(1X,12,10X,F10.4,13X,F10.4,13X,F10.4)
37 CONTINUE
40 CALL DATSW(12,KEY13)
GO TO(41,50),KEY13
C*****
C***** PRINT OUT ELECTRICAL DATA
C*****
41 WRITE(3,42)
42 FORMAT(//,1X,'ELECTRICAL DATA')
WRITE(3,43) RV,RG,TRNEG,ZGP
43 FORMAT(//,1X,RV =,F10.6,' RG =,F10.6,' TRNEG=,F10.1,' ZGP
1=,F10.2)
WRITE(3,44) REG,TRNFM,ZMP,RFM
44 FORMAT(1X,REG =,F10.6,' TRNFM=,F10.2,' ZMP =,F10.2,' RFM =,
1,F10.6)
WRITE(3,45) FVAX,C1,XK1
45 FORMAT(1X,FVAX=,F10.4,' C1 =,F10.4,' XK1 =,F10.4)
WRITE(3,46) CSHER,TAUR,FVIRV,TAUA
46 FORMAT(1X,CSHER=,F10.4,' TAUR =,F10.4,' FVIRV=,F10.4,' TAUA=,
1,F10.4)
WRITE(3,47) TRNSF,RX,XGEN,XLA,SPGEN,SPVOT
47 FORMAT(1X,TRNSF=,F10.4,' RX =,F10.4,' XGEN =,F10.4,' XLA =,
1,F10.4,' SGEN=,F10.4,' SPVOT=,F10.4)
WRITE(3,48)
48 FORMAT(1X,'CFGA',16X,'CFVA')
DO 49 I=1,13
WRITE(3,49) CFGA(I),CFVA(I)
49 FORMAT(1X,F10.4,10X,F10.4)
50 CONTINUE
50 CALL DATSW(14,L)
IF(L=1) GO TO 50
C*****
C***** PRINT OUT GOVERNOR AND DIESEL DATA
C*****
51 WRITE(3,52)
52 FORMAT(//,1X,'GOVERNOR AND DIESEL DATA')
WRITE(3,53) ALP1,ALP2,ALP3,ALPD
53 FORMAT(//,1X,ALP1 =,F10.4,' ALP2=,F10.4,' ALP3=,F10.4,' ALPD=
1,F10.4)
WRITE(3,54) XK6,QPRG,CWM
54 FORMAT(1X,XK6 =,F10.2,' QPRG=,F10.4,' CWM =,F10.4)
60 CALL DATSW(15,M)
IF(M=1) GO TO 61,70
C*****
C***** PRINT OUT SYSTEMS INERTIAS
C*****
61 WRITE(3,62)
62 FORMAT(//,1X,'SYSTEM INERTIAS')
WRITE(3,63) XJPA,XJM,XJSH
63 FORMAT(1X,XJPA=,F10.2,' XJM=,F10.2,' XJSH=,F10.2)
WRITE(3,64) XJD,XJG,XJPE
64 FORMAT(1X,XJD=,F10.2,' XJG=,F10.2,' XJPE=,F10.2)
70 CALL DATSW(10,KEY10)

```

GEN11620
 GEN11630
 GEN11640
 GEN11650
 GEN11660
 GEN11670
 GEN11680
 GEN11690
 GEN11700
 GEN11710
 GEN11720
 GEN11730
 GEN11740
 GEN11750
 GEN11760
 GEN11770
 GEN11780
 GEN11790
 GEN11800
 GEN11810
 GEN11820
 GEN11830
 GEN11840
 GEN11850
 GEN11860
 GEN11870
 GEN11880
 GEN11890
 GEN11900
 GEN11910
 GEN11920
 GEN11930
 GEN11940
 GEN11950
 GEN11960
 GEN11970
 GEN11980
 GEN11990
 GEN12000
 GEN12010
 GEN12020
 GEN12030
 GEN12040
 GEN12050
 GEN12060
 GEN12070
 GEN12080
 GEN12090
 GEN12100
 GEN12110
 GEN12120
 GEN12130
 GEN12140
 GEN12150


```

GO TO (91,99), KEY10
C*****
C***** PRINT ICE DATA
C*****
91 WRITE(3,92)
92 FORMAT(//,1X,'ICE DATA')
93 WRITE(3,93) BLOC1,D8B12,BLOC2,D8B21
94 FORMAT(//,1X,'BLOC1=',F10.3,' D8B12=',F10.3,' BLOC2=',F10.3,' D8B21=
   1=F10.3)
95 WRITE(3,94)
96 FORMAT(6X,'RBA1')
97 DO 95 I=1,25
98 WRITE(3,96) RBA11)
99 FORMAT(1X,F10.3)
99 CONTINUE
97 WRITE(3,97) DCR,D8B,CICE
98 FORMAT(1X,'DCR=',F10.3,' DSM=',F10.3,' CICE=',F10.3)
99 WRITE(3,98) CK1,CK2,CK3
99 FORMAT(1X,'CK1=',F10.3,' CK2=',F10.3,' CK3 =',F10.3)
99 CONTINUE
99 WRITE(3,71)
71 FORMAT(1M1,55X,'OUTPUT DATA',//)
C*****
C***** BEGIN SIMULTAION
C*****
CALL LINK(EXEC)
END
// DUP
*STORE WS UA GEN:5

```

```

GEN:12160
GEN:12170
GEN:12180
GEN:12190
GEN:12200
GEN:12210
GEN:12220
GEN:12230
GEN:12240
GEN:12250
GEN:12260
GEN:12270
GEN:12280
GEN:12290
GEN:12300
GEN:12310
GEN:12320
GEN:12330
GEN:12340
GEN:12350
GEN:12360
GEN:12370
GEN:12380
GEN:12390
GEN:12400
GEN:12410
GEN:12420
GEN:12430
GEN:12440

```



```

C***** HULL DATA
C*****
C      READ (2,203) CAD,DELTA,XXK
C      READ DATA CARDS 51 - 70
C      DO 102 I=1,20
C      102 READ(2,204) EHP(I),WADED(I),THDED(I)
C*****
C***** ELECTRICAL DATA
C*****
C      READ DATA CARDS 71 AND 72
C      READ(2,4000) RM,TRNFM,ZVP,REV,RG,TRNFG,ZGP,RFG
C      READ DATA CARD 73
C      READ(2,203) FIVAX,C1,XX1,CSHF,TRNSF,RX
C      READ DATA CARD 74
C      READ(2,203) FMIRV,TAUA,TAUB
C      READ DATA CARD 75
C      READ(2,2066) XGEN,XLA
C      READ DATA CARD 76
C      READ(2,203) SPGEN,SPVOT
C      READ DATA CARDS 77 - 89
C      DO 103 I=1,13
C      103 READ(2,207) CFGA(I),CFVA(I)
C*****
C***** GOVERNOR - DIESEL DATA
C*****
C      READ DATA CARD 90
C      READ(2,203) ALP1,ALP2,ALP3,ALPD,XX6,Q3RG,CWM
C*****
C***** SYSTEMS INERTIAS
C*****
C      READ DATA CARD 91
C      READ(2,203) XJPA,XJM,XJSH,XJD,XJG
C*****
C***** ICE CONDITIONS
C*****
C      READ DATA CARDS 92 - 96
C      DO 104 I=1,25,5
C      104 READ(2,230) RBA(I),RBA(I+1),RBA(I+2),RBA(I+3),RBA(I+4)
C      READ DATA CARD 97
C      READ(2,230) BLOC1,BRR12,BLOC2,BBR21,CICE,DCR,DSH
C*****
C***** SCALE FACTORS, INITIAL CONDITIONS, CONTROL DATA
C*****
C      READ DATA CARD 98
C      READ(2,214) SY1,SY2,SY3,SY4,SY5,SY6,SY7
C      READ DATA CARD 99
C      READ(2,214) SY8,SY9,SY10,SY11,SY12,SY13
C      READ DATA CARD 100
C      READ(2,203) T,THEIC,XNPM,Y,Z,IRV,TEMP
C      READ DATA CARD 101
C      READ(2,203) FGI,XNPM,FIP,XNP,V5,XS
C      READ DATA CARD 102
C      READ(2,203) AI,AID0

```

REDA0540
 REDA0550
 REDA0560
 REDA0570
 REDA0580
 REDA0590
 REDA0600
 REDA0610
 REDA0620
 REDA0630
 REDA0640
 REDA0650
 REDA0660
 REDA0670
 REDA0680
 REDA0690
 REDA0700
 REDA0710
 REDA0720
 REDA0730
 REDA0740
 REDA0750
 REDA0760
 REDA0770
 REDA0780
 REDA0790
 REDA0800
 REDA0810
 REDA0820
 REDA0830
 REDA0840
 REDA0850
 REDA0860
 REDA0870
 REDA0880
 REDA0890
 REDA0900
 REDA0910
 REDA0920
 REDA0930
 REDA0940
 REDA0950
 REDA0960
 REDA0970
 REDA0980
 REDA0990
 REDA1000
 REDA1010
 REDA1020
 REDA1030
 REDA1040
 REDA1050
 REDA1060
 REDA1070

REDA1080
 REDA1090
 REDA1100
 REDA1110
 REDA1120
 REDA1130
 REDA1140
 REDA1150
 REDA1160
 REDA1170
 REDA1180
 REDA1190
 REDA1200
 REDA1210
 REDA1220
 REDA1230
 REDA1240
 REDA1250
 REDA1260

```

C      READ(2,203) THET1,TAU1,CK1,CK2,CK3
C      READ(2,204) TEND1,TAU1,CK1,CK2,CK3
C      READ(2,205) ICNTR,NON,DT,TIMSC,TICE,TEND
C
C      SET CONSTANTS 'RHO' AND 'PI'
C
C      RHO=1.99
C      PI=3.14159
C
C      COMPUTE 'RA' THUS LEAVING VARIABLE 'RM' OPEN
C      FOR USE AS LOCAL VARIABLE IN SUBROUTINE QICE
C
C      COMPUTE 'MASS' TERM FOR THRUST COMPUTATION
C      XMASS=(1.+CAD)*2240.*DELTA/32.18
C
C      RETURN
C      END
C
C      // DUP
C      *STORE      WS  UA  REDAT
  
```



```

// FOR
LIST ALL
*ONE WORD INTEGERS
SUBROUTINE POLM1
VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968

COMMON PO,PI,IZB,DAR,PDR,D,XJPA,XJV,XJSH,XJD,XJG,EEG,
1 XJP,XJOG,FIVAX,PAR,RG,C1,XK1,QBRG,CSHR,CWM,PCVS,XLA,
2 TAU1,TAU2,XK6,ALP1,ALP2,ALP3,ALPD,XMASS
COMMON NOVAR,NVAR(10),JJ,ICNTR,NPLOT,NDATA,NOP,NON,NIP
COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12
COMMON DT,CT,T,TEND,TIMSC
COMMON Z,Y,X,NMD,FGI,FI,P,XNP,VS,XS,SIE,A,RV,TEMP
COMMON CZ,CY,CXNPM,CFG1,CFIP,CXNP,CVS,CXS,CSIE,CAL,CRV,CTEMP
COMMON ZDO,YDO,XNPM,D,FGIDDO,FI,PDO,XNPD,VSDO,XSDO,SIEDO,AIDO,RVDO,TPOLM0:140
1 FVDO
COMMON XNE,XNEDO,VAI,THETO,THET1,THETA,THETD,THEIC,QDE,OG,EG,EB,EC,POLM0:150
10V,CFMV,AIHAT,QP,OP,XOI,TP,XPROI,R,VAR,XKW,CVK
COMMON CFG,SPGEN,CFGA(12),TRNFG,ZGP,RFQ,XLFG,TAUFG,TRNSF,RX,XM,XLAPOLM0:160
1SF,AMPTR,XGEN
COMMON CFM,SPNOT,CFMA(13),TRNFV,ZVP,RFM,XLFM,TAUFV,FMIRV,TAUA,TAUBPOLM0:200
COMMON B(64),SB(64),E(32),EE(32)
COMMON EHP(20),WDED,XADE(20),TDDED,THDED(20),XKK
COMMON IELO,JELO,KELO,LELO,MELO
COMMON SPARE(5),ISPAR(5)

EQUIVALENCE (XJPE,PCMS)

GO TO (3,3,3,4),IZB
3 XJPE=(1.024*((1.974DAR*PDR)-9.30)*1D/1.33)**5)/32.18
GO TO 20
4 XJPE=(1.024*((1.094DAR*PDR)-0.23)*1D/1.33)**5)/32.18
20 IF (XJPA) 10,5,10
5 XJPA=52.65**2
10 XJNP=XJV+XJSH+XJPA+XJPE
XJOG=(XJD+XJG)*XGEN
RETURN
END

// NUP
STORE WS UA POLM1

```



```

// FOR
*LIST ALL
*NAME EXEC
*ONE WORD INTEGERS
*IOCS(CARD,TYPEWRITER,KEYBOARD,1132PRINTER,DISK)
C ICERBREAKER PROPULSION PLANT SIMULATION
C EXECUTIVE ROUTINE
C
C VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968
C
COMMON RO,PI,IZB,DAR,PDR,D,XUPA,XUM,XUS4,XJD,XJG,EPG,
1 XJVP,XJUG,FINAX,PA,PG,CI,XK1,CBRG,CSHER,CWM,PCMS,XLA,
2 TAU1,TAU2,XK6,ALP1,ALP2,ALP3,ALP4,XMA55
COMMON NOVAR,NVAR(10),JJ,ICNTR,NPLOT,NDATA,NOP,NON,NIP
COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12
COMMON DT,CT,T,TEND,TIMSC
COMMON Z,Y,X,NV,FGI,FIP,XNP,VS,XS,SIE,AI,RV,TEMP
COMMON CZ,CY,CX,NP,CTG1,CFIP,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP
COMMON ZDO,YDO,XNP,D,FGIDO,FIPDO,XNPDO,VSDO,XSDO,SIEDO,AIDO,RVDO,T
1 ENDO
COMMON XNS,XNEDO,VA,THETO,THET1,THETA,THETD,THEIC,GDE,CG,EG,EB,EC,
COMMON AINHAT,OP1,OP1,TP,XPROI,R,VAR,XKM,CVK
COMMON CFG,SPGEN,CFGA(13),TRNFG,ZGP,REF,XLFG,TAUFG,TRNSF,RX,XV,XLA
1 SE,ANDTR,XGEN
COMMON CFM,SPNOT,CFVA(13),TRNFM,ZVP,REF,XLFM,TAUFM,FYIRV,TAUA,TAUB
COMMON B(64),BB(64),E(132),EE(32)
COMMON EHP(20),XQED,XQED(20),TDDE,THDED(20),XKK
COMMON IELO,JELO,XELO,LELO,NELO
COMMON SPARE(5),ISPAR(5)
COMMON CAD,DELTA,RV
COMMON RBA(25),PRVX,BLOC1,BB12,BLOC2,DBB21
COMMON FO,EK,DCI,DCR,DSH,CICE,TICE
COMMON CK1,CK2,CK3
EQUIVALENCE (FMI,SPARE(1))
SPARE(1)--FV:
SPARE(2)--XNGR
SPARE(3)--XNGD
SPARE(4)--VACANT
SPARE(5)--VACANT
ISPAR(1)--CONTROLS CALLING OF GENIS FROM EXEC
ISPAR(2)--CONTROLS MOTOR FIELD PLUGGING IN CALVR
ISPAR(3)--VACANT
ISPAR(4)--VACANT
ISPAR(5)--VACANT
DEFINE FILE 11600,26,U,IELO)
INITIALIZE CONTROL VARIABLES
SET DISK WRITE TIME STEP CONTROL
III=1
MIKE=1
SET PRINTER STEP COUNTER
SET DISK FILE INDEX
EXEC 00
EXEC 010
EXEC 020
EXEC 030
EXEC 040
EXEC 050
EXEC 060
EXEC 070
EXEC 080
EXEC 090
EXEC 100
EXEC 110
EXEC 120
EXEC 130
EXEC 140
EXEC 150
EXEC 160
EXEC 170
EXEC 180
EXEC 190
EXEC 200
EXEC 210
EXEC 220
EXEC 230
EXEC 240
EXEC 250
EXEC 260
EXEC 270
EXEC 280
EXEC 290
EXEC 300
EXEC 310
EXEC 320
EXEC 330
EXEC 340
EXEC 350
EXEC 360
EXEC 370
EXEC 380
EXEC 390
EXEC 400
EXEC 410
EXEC 420
EXEC 430
EXEC 440
EXEC 450
EXEC 460
EXEC 470
EXEC 480
EXEC 490
EXEC 500
EXEC 510
EXEC 520
EXEC 530

```



```

C      JELO=1      SET ICE CONTROL
C      JELO=1
C      KELO=1      SET MOTOR FIELD PLUG
C      ISPAR(2)=1  SAVE DISK WRITE TIME STEP
C      DTT=TIMSC   COMPUTE NEXT TIME FOR WRITING ON DISK
C      TTT=TT+TIMSC
C      COMPUTE INITIAL VALUES
C      CALL CNTRL
C      PRINT OUT INITIAL VALUES
C      CALL OUT
C      CHECK IF DATA SWITCHES ARE TO BE INTERROGATED
C      CALL DATSW (0,KEYO)
C      GO TO (40,1300), KEYO
C      CALL NUMERICAL INTEGRATION ROUTINE TO UPDATE VALUES
C      1300 CALL HEUN
C      CHECK FOR PRINTING OUT OF DATA
C      2 IF (VIKE=NON) 31,92,92
C      32 CALL OUT
C      VIKE=0
C      31 VIKE=VIKE+1
C      CHECK FOR END OF SIMULATION
C      IF (CT-(TTEND+DTT)) 51,51,40
C      CHANGE DISK WRITE INTERVAL IF TIME GREATER THAN 20 SECONDS
C      51 IF (CT-20.) 30,30,10
C      10 GO TO (11,30),111
C      11 DTT=1.0
C      TTT=TTT+DTT-TIMSC
C      111=2
C      30 IF (CT-TTT) 50,55,55
C      CONVERT VALUES TO UNITS FOR WRITING ON DISK
C      55 RPNP=60.*XNDV
C      RPNP=60.*XND
C      CVK=CVS/1.689
C      XKN=.745*CV+2.*OI*XND/550.
C      WRITE VALUES ON DISK
C      WRITE (1,1E10) CT,AI,EB,RPNP,XKN,CVK,OPI,XPROI,CXS
C      1,01,TEVP
C      UPDATE TIME FOR NEXT DISK WRITING
C      TTT=TTT+DTT
C      GO TO 50
C      INTERROGATE DATA SWITCHES
C      40 CALL FINIS
C      BEGIN NEW SIMULATION IF SO DIRECTED
C      JENE=ISPAR(1)
C      GO TO (1300,45), JENE
C      45 CALL LINK(GENIS)
C      END
C      // DUP
C      *STORE      WS   UA   EXEC

```

```

EXEC0540
EXEC0550
EXEC0560
EXEC0570
EXEC0580
EXEC0590
EXEC0600
EXEC0610
EXEC0620
EXEC0630
EXEC0640
EXEC0650
EXEC0660
EXEC0670
EXEC0680
EXEC0690
EXEC0700
EXEC0710
EXEC0720
EXEC0730
EXEC0740
EXEC0750
EXEC0760
EXEC0770
EXEC0780
EXEC0790
EXEC0800
EXEC0810
EXEC0820
EXEC0830
EXEC0840
EXEC0850
EXEC0860
EXEC0870
EXEC0880
EXEC0890
EXEC0900
EXEC0910
EXEC0920
EXEC0930
EXEC0940
EXEC0950
EXEC0960
EXEC0970
EXEC0980
EXEC0990
EXEC1000
EXEC1010
EXEC1020
EXEC1030
EXEC1040
EXEC1050
EXEC1060
EXEC1070

```



```

C      COMPUTE INITIAL VALUES OF VARIABLES
C      CALL CALVR
C      SAVE INITIAL VALUES AS CURRENT VALUES
CT=T
CAI=AI
CY=Y
CZ=Z
CXNPM=XNPM
CFIP=FI
CFG1=FGI
CXNP=XNP
CVS=VS
CXS=XS
CSIE=SIE
C      PRINT COLUMN HEADINGS FOR OUTPUT DATA
1 WRITE (3,201)
201 FORMAT (1X,'T(SEC)  RPM  VK  ODE(FT-LB)  QG  EB(V)')
1 EVI EC OM(FT-LB) AI(A) OP THRUST(T) RESIST HP/SH')
C      CONVERT TO APPROPRIATE UNITS FOR DISK WRITING
RPM=60.*XNP
VK=CVS/1.689
XKW=.746*OM*2.*PI*XNP/550.
C      WRITE INITIAL VALUES ON DISK
WRITE (1,1E10) CT,AI,ER,RPM,FM,RPMP,XKW,CVK,OP,XPROI,CXS
1.01,TEMP
RETURN
END
// SUP
*STORE WS UA CNTRL

```

```

CNTR0540
CNTR0550
CNTR0560
CNTR0570
CNTR0580
CNTR0590
CNTR0600
CNTR0610
CNTR0620
CNTR0630
CNTR0640
CNTR0650
CNTR0660
CNTR0670
CNTR0680
CNTR0690
CNTR0700
CNTR0710
CNTR0720
CNTR0730
CNTR0740
CNTR0750
CNTR0760
CNTR0770
CNTR0780
CNTR0790
CNTR0800
CNTR0810
CNTR0820
CNTR0830

```



```

// FOR
*LIST ALL
*ONE WORD INTEGERS
SUBROUTINE HEUN
C
VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968
DIMENSION STVR(12), CSTVR(12), STVRD(12), STVR0(12), STVR1(12), STVR2
117)
COMMON RO,P1,I2R,DAR,PDR,D,XJPA,XJM,XUSH,XJD,XJG,EFQ,
1 XJNP,XJGG,F1MAX,RA,RG,C1,XK1,CBRG,COPER,CWM,PCYS,XLA,
2 TAU1,TAU2,XK3,ALP1,ALP2,ALP3,ALP4,XWASS
COMMON NOVAR,NVAR(10),JJ,ICNTR,NPLOT,NDATA,NOP,NOV,NIP
COMMON SX,SY,SV1,SV2,SV3,SV4,SV5,SV6,SV7,SV8,SV9,SV10,SV11,SV12
COMMON DT,CT,T,TEND,TIMSC
COMMON Z,Y,XNPV,FG,F,P,XNP,VS,XS,SIE,AI,RV,TEVP
COMMON CZ,CY,CXNP,CFG,CFIP,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEXP
COMMON ZDO,YDO,XNPVD,FGIDO,FIPDO,XNPDO,VSDO,XSDO,SIEDO,AIDO,RVDO,T
1 FMDO
COMMON XNE,XNEDO,VA,THETO,THET1,THETA,THETD,THEIC,ODE,OG,EG,EB,EC,
1 OM,OFER,AIMAT,OP,QPI,QI,QI,TP,XPROI,R,VAR,XKW,CVK
COMMON CFG,SPGEN,CFGGA(13),TRNFG,ZGP,REF,XLFG,TAUFG,TRNSF,RX,XV,XLA
1 SF,ANSTR,XGEN
COMMON CFV,SPVOT,CFVA(13),TRNFV,ZMP,RFM,XLFV,TAUFM,FMIRV,TAUA,TAUS
COMMON B(64),B(164),E(32),EE(32)
COMMON EMP(20),NDEP,XADE(20),TDED,THDED(20),XKK
COMMON IELD,JELD,XELO,XELO,XELO
COMMON SPARE(5),ISPAR(5)
EQUIVALENCE (Z,STVR(1))
EQUIVALENCE (CZ,CSTVR(1))
EQUIVALENCE (ZDO,STVRD(1))
CSTVR ARRAY REPRESENTS OLD VALUES
STVR ARRAY REPRESENTS DERIVATIVES
STVR0 ARRAY REPRESENTS CURRENT SOLUTION IN INTEGRATION, UPDATED
STVR1 ARRAY WILL CONTAIN VALUES OF FIRST ORDER
STVR2 ARRAY WILL CONTAIN VALUES OF SECOND ORDER
STVR3 ARRAY WILL CONTAIN VALUES OF THIRD ORDER
C
AS EACH NEW STEP IS CALCULATED
N=12
DO 20 I=1,N
DO 20 I=1,N
STVR0(I)=STVRD(I)*DT
10 STVR(I)=CSTVR(I)+STVR0(I)/2.0
IJAX=1
GO TO 100
200 T=CT+DT
CALL CALVR
DO 20 I=1,N
STVR1(I)=STVRD(I)*DT
20 STVR(I)=CSTVR(I)+STVR1(I)/2.0
IJAX=2
GO TO 100
300 T=CT+DT
CALL CALVR
DO 20 I=1,N

```



```

30 STVR(I)=STVR(I)*DT
   STVR(I)=CSTVR(I)+STVR2(I)
   IJAX=3
   GO TO 100
400 T=CT+DT
   CALL CALVR
   DO 40 I=1,N
40 STVR(I)=CSTVR(I)+(STVR(I)+2.0*(STVR1(I)+STVR2(I))+STVRD(I)*DT)/6.
   IJAX=4
   C   RESET SERVO STROKE POSITION TO ZERO IF LESS THAN ZERO OR TO 0.6 IF
   C   GREATER THAN 0.6
100 IF(Z) 162,163,163
162 Z=0.
   GO TO 16
163 IF Z-1.0 16,16,160
160 Z=1.0
   C   SET MOTOR FIELD ADJUSTING CURRENT TO ZERO IF NEGATIVE
16 IF(FIP) 172,172,173
172 FIP=0.0
   C   CONTINUE WITH SOLUTION
173 GO TO (200,300,400,500), IJAX
   C   SAVE NEW VALUES AS CURRENT VALUES
500 DO 50 I=1,N
50 CSTVR(I)=STVR(I)
   CT=T
   C   UPDATE VALUES AND COMPUTE DERIVATIVES
   CALL CALVR
   RETURN
   END
// DUP
*STORE      WS  UA  HEUN

```

HEUN0530
 HEUN0540
 HEUN0550
 HEUN0560
 HEUN0570
 HEUN0580
 HEUN0590
 HEUN0600
 HEUN0610
 HEUN0620
 HEUN0630
 HEUN0640
 HEUN0650
 HEUN0660
 HEUN0670
 HEUN0680
 HEUN0690
 HEUN0700
 HEUN0710
 HEUN0720
 HEUN0730
 HEUN0740
 HEUN0750
 HEUN0760
 HEUN0770
 HEUN0780
 HEUN0790
 HEUN0800
 HEUN0810
 HEUN0820
 HEUN0830

CALV0340
CALV0350
CALV0360
CALV0370
CALV0380
CALV0390
CALV0400
CALV0410
CALV0420
CALV0430
CALV0440
CALV0450
CALV0460
CALV0470
CALV0480
CALV0490
CALV0500
CALV0510
CALV0520
CALV0530
CALV0540
CALV0550
CALV0560
CALV0570
CALV0580
CALV0590
CALV0600
CALV0610
CALV0620
CALV0630
CALV0640
CALV0650
CALV0660
CALV0670
CALV0680
CALV0690
CALV0700
CALV0710
CALV0720
CALV0730
CALV0740
CALV0750
CALV0760
CALV0770
CALV0780
CALV0790
CALV0800
CALV0810
CALV0820
CALV0830
CALV0840
CALV0850
CALV0860
CALV0870
CALV0880
CALV0890
CALV0900
CALV0910
CALV0920
CALV0930
CALV0940
CALV0950
CALV0960
CALV0970
CALV0980
CALV0990
CALV1000
CALV1010
CALV1020
CALV1030
CALV1040
CALV1050
CALV1060
CALV1070

CALV1080
CALV1090

// DUP
#STORE WS UA CALVR


```

// FOR
*LIST ALL
*ONE WORD INTEGERS
SUBROUTINE CON1
C
C   VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968
C
COMMON RO,PI,IZB,DAR,DDR,D,XJPA,XJM,XJSH,XJD,XJG,EFG,
1 XJWP,XJUG,FIVAX,RA,RG,C1,XK1,QBRG,CSHR,CWM,PCMS,XLA,
2 TAU1,TAU2,XK6,ALP1,ALP2,ALP3,ALPD,XNASS
COMMON NOVAR,NVAR(10),JJ,ICNTR,NPLOT,NDATA,NOP,NON,NIP
COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12
COMMON DT,CT,Y,TEND,TINSC
COMMON Z,Y,XNEM,FGI,FIP,XNP,V5,XS,SIE,AI,RV,TEVP
COMMON CZ,CY,CXNP,CY,CXNP,CY,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP
COMMON ZDO,YDO,XNPD,FGID,FGID,FIPDO,XNPD,VSDO,XSDO,SIEDO,AIDO,RVDO,TC
1 EWDG
COMMON XNE,XNEDO,VA,THETO,THET1,THETA,THETD,THEIC,GDE,OG,EG,EB,EC,
107,CFRM,AIHAT,GP,CPI,GI,TP,XPROI,R,VAR,XKX,CVK
COMMON CFG,SPGEN,CFGA(13),TRNG,ZG,REG,XLFG,TAUFG,TRNSF,RX,XX,XLAC
15F,AVPTR,XGEN
COMMON CFV,SPNOT,CFMA(13),TRNG,ZMP,REM,XLFM,TAUFV,FMIRV,TAUA,TAUS
COMMON B(64),BB(64),EE(32)
COMMON EHP(20),WDED,WADED(20),TDDE,THDED(20),XKK
COMMON JELO,JELO,KELO,LELO,VELO
COMMON SPARE(5),ISPAR(5)
C
C   CHECK FOR RANGE ZERO TO TAU1 SECONDS
C
C   IF (T-TAU1) 5,5,10
C
C   TIME LESS THAN TAU1 SECONDS
C
C   5 THETA=THEIC+(THET1-THETC)*T/TAU1
C   THETD=(THET1-THETC)/TAU1
C   GO TO 25
C
C   CHECK FOR RANGE TAU1 TO TAU2 SECONDS
C
C   10 IF (T-(TAU1+TAU2)) 15,20,20
C
C   TIME BETWEEN TAU1 AND TAU2 SECONDS
C
C   15 THETA=THET1+(THETO-THET1)*(T-TAU1)/TAU2
C   THETD=(THETO-THET1)/TAU2
C   GO TO 25
C
C   TIME GREATER THAN OR EQUAL TO TAU2 SECONDS
C
C   20 THETA=THETO
C   THETD=0.0
C   25 RETURN
C   END
// DUP
*STORE WS UA CON1

```


C20N0540
C20N0550

// SUP
*STORE WS UA CONZ


```

// FOR
*LIST ALL
*ONE WORD INTEGERS
SUBROUTINE FLUXG
C
C   VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968
C
COMMON RO,PI,IZE,DAR,PDR,D,XJPA,XJV,XJSH,XJD,XJG,EFG,
1 XJVP,XJOG,FIVAX,RA,RG,CL,XK1,OBGR,CSHR,CNM,PCMS,XLA,
2 TAU1,TAU2,XK6,ALP1,ALP2,ALP3,ALP4,XVASS
COMMON NOVAR,NVAR(10),J,JCNTR,NPLOT,NDATA,NOP,NON,NIP
COMMON SX,SY,SY1,SY2,SY3,SY4,SV5,SY6,SY7,SY8,SY9,SY10,SY11,SY12
COMMON DT,CT,T,TEND,TINSC
COMMON Z,Y,X,NPH,FGL,FIP,XNP,V5,XS,SIE,AI,RV,TEMP
COMMON CZ,CV,CKUPY,CFGI,CFIP,CKNP,CVS,CXS,CSIE,CAI,CRV,CTEMP
COMMON ZDO,YDO,XNPVD,FGIDDO,FIPDO,XNPDO,VSDO,XSDO,SIEDO,AIDO,RVDO,TE
1 FVDO
COMMON XNE,XNEDO,VA,THETO,THET1,THETA,THETD,THEIC,ODE,OG,EG,EB,EC,
10V,GERV,AIHAT,OP,CPI,QI,TP,XPROI,R,VAR,XKX,CVK
COMMON CFG,SPGEN,CFGAL(13),TRNEG,ZGP,RFG,XLFG,TAUFG,TRNSF,RX,XY,XLAF
15F,AMPTR,XGEN
COMMON CFM,SPNOT,CFMA(13),TRNEV,ZNP,RFM,XLFM,TAUFM,FYIRV,TAUA,TAUB
COMMON E(64),RE(64),F(32),EE(32)
COMMON EPL(20),WDED,XADED(20),TDED,TXDED(20),XKK
COMMON IFLO,JELO,XELO,VELO,YELO
COMMON SPARE(5),ISPAR(5)
C
C   COMPUTE ABSCISSA
FG=ABS(AMPTR)/SPGEN
C
C   TRUNCATE TO INTEGER
KG=FG
C
C
C   CHECK IF ARSCISSA EQUAL TO LAST VALUE, AND IF
C   SO SET ORIGINATE TO LAST VALUE AND INTERVAL TO
C   LAST RANGE
IF (KG-11) 1,1,2
2 CFG=CFGAL(13)
DCGD1=CFGAL(13)-CFGAL(12)
GO TO 3
C
C   COMPUTE RANGE ORIGINATE FALLS IN
1 DCGDI=CFGAL(KG+2)-CFGAL(KG+1)
C   COMPUTE ORIGINATE
CFG=CFGAL(KG+1)+DCGDI*(FGL-KG)
C   SET SIGN OF ORIGINATE
3 CFG=AVPTR/ABS(AMPTR)*CFG
C
C   COMPUTE REMAINING GENERATOR VALUES
XLFG=TRNEG*TRNEG*DCGDI/(ZGP*SPGEN)
TAUFG=XLFG/(RX+RG)
XM=TRNEG*TRNSF*DCGDI/(ZGP*SPGEN)
XLASF=TRNSF*TRNSF*DCGDI/(ZGP*SPGEN)
RETURN
END
// DUP
*STORE WS UA FLUXG

```

```

FLXG0000
FLXG0010
FLXG0020
FLXG0030
FLXG0040
FLXG0050
FLXG0060
FLXG0070
FLXG0080
FLXG0090
FLXG0100
FLXG0110
FLXG0120
FLXG0130
FLXG0140
FLXG0150
FLXG0160
FLXG0170
FLXG0180
FLXG0190
FLXG0200
FLXG0210
FLXG0220
FLXG0230
FLXG0240
FLXG0250
FLXG0260
FLXG0270
FLXG0280
FLXG0290
FLXG0300
FLXG0310
FLXG0320
FLXG0330
FLXG0340
FLXG0350
FLXG0360
FLXG0370
FLXG0380
FLXG0390
FLXG0400
FLXG0410
FLXG0420
FLXG0430
FLXG0440
FLXG0450
FLXG0460
FLXG0470
FLXG0480
FLXG0490
FLXG0500
FLXG0510

```



```

01CE0000
01CE0010
01CE0020
01CE0030
01CE0040
01CE0050
01CE0060
01CE0070
01CE0080
01CE0090
01CE0100
01CE0110
01CE0120
01CE0130
01CE0140
01CE0150
01CE0160
01CE0170
01CE0180
01CE0190
01CE0200
01CE0210
01CE0220
01CE0230
01CE0240
01CE0250
01CE0260
01CE0270
01CE0280
01CE0290
01CE0300
01CE0310
01CE0320
01CE0330
01CE0340
01CE0350
01CE0360
01CE0370
01CE0380
01CE0390
01CE0400
01CE0410
01CE0420
01CE0430
01CE0440
01CE0450
01CE0460
01CE0470
01CE0480
01CE0490
01CE0500
01CE0510
01CE0520

// FOR
*LIST ALL
*ONE WORD INTEGERS
SUBROUTINE OICE
VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968

COMMON RO,PI,IZB,DAR,ODR,D,XJPA,XJM,XJSH,XJD,XJG,EEG,
1 XJPD,XJGG,FIVAX,RAVGO,CI,XK1,QBRG,CSHER,CWM,PCMS,XLA,
2 TAU1,TAU2,XK6,ALP1ALP2,ALD3,ALPD,XVASS
COMMON NOVAR,NVAR(10),JJ,IGNR,NLOC,NDATA,NOP,NON,NIP
COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12
COMMON DT,CT,T,TEND,TMSC
COMMON Z,Y,X,NON,FGI,FIP,XNP,VS,XS,SIE,AI,RV,TEMP
COMMON CZ,CY,CX,NPM,CFG,CFIP,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP
COMMON ZDO,VDO,XNPV,FGID,FIPO,XNPDO,VSDO,XSDO,SIEDO,AIDO,RVDO,TO
1 EVD0
COMMON XVE,XNEDO,VA,THETO,THET1,THETA,THETD,THEIC,ODE,GG,EG,E8,EC,Q
10V,XFERM,AIHAT,GP,CPI,OI,TP,XPROI,R,VAR,XKW,CVK
COMMON CFG,SPGEN,CFGA113,TRNFG,ZSP,XFG,XLFG,TAUFG,TRNCF,RX,XM,XLAC
1 SE,XNSTR,XGEN
COMMON CFY,SPVOT,CFYA(13),TRNFM,ZXP,RFM,XLFM,TAUFY,FMIRV,TAUA,TAUBQ
COMMON P(64),PB(64),E(32),EE(32)
COMMON EWP(20),WDED,WADED(20),TDED,THDED(20),XKK
COMMON IELO,JELO,KELO,LELO,VELO
COMMON SPARE(5),ISPAR(5)
COMMON CAD,DELTA,RV
COMMON RBA(25),REXX,BLOC1,DBB12,BLOC2,DBB21
COMMON EO,EK,DCT,DCR,DSH,CICE,TICE
COMMON CK1,CK2,CK3

CHECK FOR INITIAL CALCULATION OF MAX CUT

GO TO (1,2), JELO
CALL RRVAX

CALCULATE ANGLE OF CUT

CSAB2=1-(2.*R9MX/D)
SNAB2=SQRT(1.-CSAB2**2)
AH2=2.*ATAN(SNAB2/CSAB2)

CALCULATE ANGLE BETWEEN PROP BLADES

XIZR=IZB
AMD=2.*PI/XIZB
JELO=2

CHECK IF ANGULAR POSITION OF PROP LESS THAN
OR EQUAL TO ANGLE OF CUT

2 IF(SIE-AB2) 30,30,20

CHECK IF ANGULAR POSITION OF PROP GREATER
THAN OR EQUAL TO ANGLE BETWEEN PROP BLADES

20 IF(SIE-AB8) 25,1,1

SET ICE TORQUE TO ZERO

25 OI=0.
RETURN

CHECK IF PROP ANGULAR VELOCITY BELOW 0.015

30 IF(ABS(XNP)-.015) 25,25,31
31 CSBET=CSAB2*COS(SIE)-SNAB2*SIN(SIE)

```


GICE0530
 GICE0540
 GICE0550
 GICE0560
 GICE0570
 GICE0580
 GICE0590
 GICE0600
 GICE0610
 GICE0620
 GICE0630
 GICE0640
 GICE0650
 GICE0660
 GICE0670
 GICE0680
 GICE0690
 GICE0700
 GICE0710
 GICE0720
 GICE0730
 GICE0740

```

      RB=N/2*(1.-CSAB2/CSBET)
      RV=(D-R)/2.
      ALFAV=ATAN(VS/(RM*2.*PI*XNP))
      ALFAI=PDR*D/(2.*PI*RV)
      EI=E0+RB*(EK-E0)/(D-DCT))
      BAV=DAR*PI*D**2/(2.*XIZB*(D-DCT))
      C      CALCULATE CRUSHING-SHEARING ICE TORQUE
      GICRS=DCR*EI*RB*RM+DSH*RB*RM*VS/(XIZB*XNP)
      C      CALCULATE MINIMUM ICE TORQUE
      GIVIN=DCR*EI*RB*RM
      C      CALCULATE CRUSHING ICE TORQUE
      GICR=DCR*BAV*RB*RM*ABS(SIN(ALFAI-ALFAV))
      C      SET ICE TORQUE
      GICE0630
      IF(GI-GICR) 45,45,40
      GICE0640
      IF(GI-GIVIN) 50,50,55
      GICE0650
      RETURN
      END
      // RUP
      *STORE      WS  UA  GICE
  
```



```

// FOR
*LIST ALL
*ONE WORD INTEGERS
SUBROUTINE RBMAX
C
VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968
COMMON RO,P1,ZB,DAR,PDR,D,XJPA,XJM,XJSH,XJD,XJG,EFQ,
1 XJVP,XJGG,FXAX,RA,RG,CI,XKI,QBRG,CSHR,CWM,PCMS,XLA,
2 TAU1,TAU2,XK6,ALP1,ALP2,ALP3,ALPD,XWASS
COMMON NOVAR,NVAR(10),JJ,ICNTR,NPLOT,NDATA,NOP,NON,NIP
COMMON SX,SY,SV1,SY2,SY3,SV4,SV5,SV6,SV7,SV8,SV9,SY10,SY11,SY12
COMMON DT,CT,T,TEND,TIMSC
COMMON ZV,XNPM,FGI,FIP,XNP,VS,XS,SIE,AI,RV,TEVP
COMMON CZ,CY,CXNPM,CFG1,CFG2,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP
COMMON ZDO,YDO,XNPD,FGIDQ,FIDQ,XNPD,VSDO,XSDO,SIEDO,AIDQ,RVDO,TRVVA0130
1 FVDO
COMMON XNE,XNEDO,VA,THETO,THET1,THET2,THETA,THETD,THEIC,QDE,OG,EG,EB,EC,RBVA0150
10V,GERM,AIHAT,OP,GPI,QI,TP,XPROI,R,VAR,XKW,CVK
COMMON CFG,SPGEN,CFGAL(13),TRNFG,ZGP,REG,XLFG,TAUFG,TRNSF,RX,XM,XLAR,SVAC0170
1SF,AUTR,XGEN
COMMON CFV,SPHOT,CFVA(13),TRNFV,ZVP,RFV,XLFV,TAUFV,FMIRV,TAUA,TAUB,SVAC0190
COMMON B(64),B3(64),E(32),EE(32)
COMMON EPR(20),WDED,WDED(20),TDDED,THDED(20),XKK
COMMON IFLO,JELO,KELO,LELO,XELO
COMMON SPARE(5),ISPAR(5)
COMMON CAD,DELTA,RV
COMMON RBA(25),RBMX,BLOC1,DB512,BLOC2,DB821
COMMON EO,EK,DCT,DCR,DSH,CICE,TICE
COMMON CK1,CK2,CK3
C
CSIE=0.
SIE=0.
C
GO TO (1,2,3,4,5), KELO
SAVE PRESENT DISTANCE
XICE=CXS
IA=1
KELO=2
C
CHECK IF IN BLOCK 1
IF((CXS-XICE)-BLOC1) 55,55,10
COMPUTE MAX CUT AND UPDATE ARRAY INDEX
RBMX=RBA(IA)*D
IA=IA+1
C
IF (IA=11) 7,7,6
IA=1
RETURN
C
REINITIALIZE ARRAY INDEX IF BEYOND MAX VALUE
UPDATE 'KELO' AND SAVE PRESENT DISTANCE
C
BRANCH ON 'KELO' AS FOLLOWS...
KELO=1 -- FIRST ENCOUNTER WITH BLOCK 1
=2 -- IN BLOCK 1
=3 -- BETWEEN BLOCKS 1 AND 2
=4 -- IN BLOCK 2
=5 -- BETWEEN BLOCKS 2 AND 1
C
GO TO (1,2,3,4,5), KELO
SAVE PRESENT DISTANCE
XICE=CXS
IA=1
KELO=2
C
CHECK IF IN BLOCK 1
IF((CXS-XICE)-BLOC1) 55,55,10
COMPUTE MAX CUT AND UPDATE ARRAY INDEX
RBMX=RBA(IA)*D
IA=IA+1
C
IF (IA=11) 7,7,6
IA=1
RETURN
C
REINITIALIZE ARRAY INDEX IF BEYOND MAX VALUE
UPDATE 'KELO' AND SAVE PRESENT DISTANCE
C

```


RBMA0540
RBMA0550
RBMA0560
RBMA0570
RBMA0580
RBMA0590
RBMA0600
RBMA0610
RBMA0620
RBMA0630
RBMA0640
RBMA0650
RBMA0660
RBMA0670
RBMA0680
RBMA0690
RBMA0700
RBMA0710
RBMA0720
RBMA0730
RBMA0740
RBMA0750
RBMA0760
RBMA0770
RBMA0780
RBMA0790
RBMA0800
RBMA0810
RBMA0820
RBMA0830
RBMA0840
RBMA0850
RBMA0860

```

10  KELO=3
    XICE=CXS
    C
    C      CHECK IF BETWEEN BLOCKS 1 AND 2 AND RETURN
    C      MAX CUT OF ZERO IF SO
    3  IF((CXS-XICE)-(DBB12) 15,15,20
    15  RBMX=0.
    RETURN
    C
    C      ENTERING BLOCK 2, INITIALIZE ARRAY INDEX AND
    C      UPDATE 'KELO'
    20  XICE=CXS
    IA=11
    KELO=4
    C
    C      CHECK IF IN BLOCK 2
    4  IF((CXS-XICE)-(BLOC2) 25,25,30
    C      COMPUTE MAX CUT AND UPDATE ARRAY INDEX
    25  RBMX=RBA(IA)*D
    IA=IA+1
    C
    C      REINITIALIZE ARRAY INDEX IF BEYOND MAX VALUE
    IF(IA-26) 27,27,26
    26  IA=1
    27  RETURN
    C
    C      BETWEEN BLOCKS 2 AND 1, UPDATE 'KELO' AND
    C      SAVE PRESENT DISTANCE
    30  KELO=5
    XICE=CXS
    C
    C      CHECK IF BETWEEN BLOCKS 2 AND 1 AND RETURN A
    C      MAX CUT OF ZERO IF SO
    5  IF((CXS-XICE)-(DBB21) 35,35,1
    35  RBMX=0.
    RETURN
    END
// DUP
*STORE WS UA RBMAX

```


STAT0530
 STAT0540
 STAT0550
 STAT0560
 STAT0570
 STAT0580
 STAT0590
 STAT0600
 STAT0610
 STAT0620
 STAT0630
 STAT0640
 STAT0650

```

C      COMPUTE PROPELLER ACCELERATION
30  XNPD0=(OM-QRM-OP-QI)/(2.*PI*XJWP)
C      COMPUTE SHIP'S ACCELERATION
C      VS00=(TP-R)/XMASS
C      XSD0=VS
C      S:ED0=XNP*2.*PI
C      RETURN
// DUP
*STORE WS UA STATE
  
```



```

// FOR
*LIST ALL
*ONE WORD INTEGERS
SUBROUTINE OUT
C
VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968
COMMON RO,PI,IZR,DAR,PDR,D,XJPA,XJM,XJSH,XJD,XJG,EFG,
1 XJW,XJG,FVAX,RA,RG,CL,XK1,CBRG,CSHR,CWX,PCMS,XLA,
2 TAU1,TAU2,XK6,ALP1,ALP2,ALP3,ALP4,XVASS
COMMON NOVAP,NVAR(10),JU,ICNTR,NPLOT,NDATA,NOP,NON,NID
COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12
COMMON DT,CT,T,TEND,TIMSC
COMMON Z,Y,XNPM,FGI,FIP,XNP,V5,XS,SIE,AI,RV,TEMP
COMMON CZ,CY,CXNPM,CFG,CFIP,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP
COMMON ZDO,YDO,XNPM,FGID,FGID,FGID,XNPD,VSDO,XSDO,SIEDO,AID,
1 RVDO,T
1,FXDO
COMMON XNE,XNDO,VA,THETO,THET1,THETA,THETD,THEIC,ODE,OG,EG,ER,EC,
1 OV,GERM,ATHAT,OP,OP1,GI,TP,XPROI,R,VAR,XKW,CKV
COMMON CFG,SPGEN,CFGA(13),TRNFG,ZCP,CFG,XLFG,TAUFG,TRNSF,RX,XY,XLA
1,SP,AVPTR,XGEN
COMMON CFM,SPVOT,CFVA(13),TRNF,ZMP,RFM,XLFM,TAUFM,FMTBV,TAUA,TAUB
COMMON B(54),BP(54),E(32),EE(32)
COMMON EBP(20),WDED,WDED(20),TDED,THDED(20),XKK
COMMON IELO,JFLO,KELO,LELO,YFLO
COMMON SPARE(5),:SPAR(5)
C
EQUIVALENCE (FV1,SPARE(1))
C
C
32 RDPV=50.*CXNPM
RDPV=50.*CXNP
C
CVK=CVS/1.689
C
TPT=TP/2240.
C
RT=R/2240.
C
C
COMPUTE SHIP'S HORSEPOWER FROM MOTOR TORQUE
AND PROPELLER ANGULAR VELOCITY
POK=GX*2.*PI*XNP/550.
PRINT VALUES
WRITE (3,301) CT,RPVPM,CFG1,RPVP,CKV,ODE,OG,EG,FM1,EG,QM,AI,OP,
1 TPT,RT,POK
301 FORMAT (1X,F7.3,1X,F5.1,1X,F5.1,1X,F6.1,1X,F5.1,1X,F8.1,
1 F8.1,1X,F6.1,1X,F5.1,1X,F6.1,1X,F9.1,1X,F3.1,1X,F9.1,1X,F7.1,1X,
2 F6.1,F8.1)
RETURN
END
// DUP
*STORE WS UA OUT

```



```

// FOR
*LIST ALL
*ONE WORD INTEGERS
C
SUBROUTINE FINIS
VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968
COMMON RO,PI,IZB,DAR,DDR,D,XJPA,XJM,XJSH,XJD,XJG,FFG,
1 XJDP,XJGG,FIWAX,RA,RG,C1,XK1,CBRG,CS,FR,CWM,PCMS,XLA,
2 TAU1,TAU2,XK5,ALP1,ALP2,ALP3,ALP4,XVASS
COMMON NOVAR,NVAR(10),JJC,ICNTR,NPLOT,NDATA,NOP,NON,NIP
COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12
COMMON DT,CT,TT,TEND,TINSC
COMMON Z,Y,XNPM,FGI,FID,XNP,V5,X5,SIE,AI,RY,TEYD
COMMON CZ,CY,CXNPM,CFG,CF,P,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP
COMMON ZDO,YDO,XNPM,D,FG,DO,FID,DO,XNPM,DO,VSDO,XSDO,SIEDO,AID,DO,RVDO,T
1 RVDO
COMMON XNE,XNEDO,VA,THETO,THET1,THETA,THETD,THEIC,ODE,OG,EG,EB,EC,FINI0150
10M,OPRM,AIHAT,OP,CPI,CI,TP,XPROI,R,VAR,XKW,CVK
COMMON CFG,SPGEN,CFGAL(13),TRNEV,ZMP,RFV,XLEN,TAUFN,FMIRV,TAUA,TAUB
15F,AVDTS,XGEN
COMMON CFY,SPVOT,CFM(13),TRNEV,ZMP,RFV,XLEN,TAUFN,FMIRV,TAUA,TAUB
COMMON B(64),B2(64),E(32),E5(22)
COMMON EDP(20),WDED,WDED(20),TDED,THDED(20),XKK
COMMON JELO,JELO,XELO,XELO,VELO
COMMON SPARE(5),SPAR(5)
CHECK IF DATA SWITCH SETTING INSTRUCTIONS
ARE TO BE TYPED OUT
C*****
C*****
CALL DATSW(4,KEY4)
GO TO(1,0,50),KEY4
50 WRITE(1,100)
100 FORMAT(' TURN ON SWITCH ---')
101 WRITE(1,101)
101 FORMAT(' 1 - TO PUNCH NEW INITIAL CONDITION CARDS')
102 WRITE(1,102)
102 FORMAT(' 2 - TO PLOT DATA')
103 WRITE(1,103)
103 FORMAT(' 3 - TO CONTINUE SOLUTION OF CURRENT PROBLEM - SWITCH OFF STARTS NEW PROBLEM')
104 WRITE(1,104)
104 FORMAT(' 4 - TO SUPPRESS FUTURE TYPING OF THESE INSTRUCTIONS')
105 WRITE(1,105)
105 FORMAT(' 5 - TO CHANGE TIME STEP')
1060 WRITE(1,1060)
1060 FORMAT(' 6 - TO CHANGE PRINT INTERVAL')
1065 WRITE(1,1065)
1065 FORMAT(' 7 - TO CHANGE XLA')
107 WRITE(1,107)
107 FORMAT(' TURN ON SELECTOR SWITCHES AND PRESS START')
110 PAUSE
CALL DATSW(1,KEY1)
GO TO(200,250),KEY1
C*****
C*****
200 WRITE(2,201) THETA,XNPM,D,Y,Z
WRITE(3,201) THETA,XNPM,D,Y,Z

```



```

201 FORMAT (10X,4F10.4,22X,'DATA0100')
WRITE(2,202) FGI,XNPM,FIP,XNP,VS,AI,AIDO
WRITE(3,202) FGI,XNPM,FIP,XNP,VS,AI,AIDO
202 FORMAT (5F10.4,22X,'DATA0101',/,2F10.4,52X,'DATA0102')
250 CALL DATSW(2,KEY2)
GO TO(300,350),KEY2
C*****
300 TEND=CT-DT
CALL LINK (GRAPH)
350 CALL DATSW(5,KEY5)
GO TO (400,450), KEY5
C*****
400 WRITE(1,120) NEW TIME STEP TO BE ENTERED
120 FORMAT(' ', ) - TIME STEP')
READ (6,121) DT
121 FORMAT(1X,F6.4)
450 CALL DATSW (6,KEY6)
GO TO (500,550), KEY6
C*****
500 WRITE (1,501) NEW PRINTER TIME STEP TO BE ENTERED
501 FORMAT (' ', ) - NON')
READ (6,502) NON
502 FORMAT (1X,I3)
550 CALL DATSW (7,KEY7)
GO TO (600,650), KEY7
C*****
600 WRITE (1,601) NEW ARMATURE CIRCUIT INDUCTANCE TO BE ENTERED
601 FORMAT(' ', ) - XLA')
READ (6,602) XLA
602 FORMAT (1X,F7.4)
C*****
STORE INFO FOR ABORTING RUN
SWITCH 3 ON TO CONTINUE, OFF TO ABORT
C*****
650 CALL DATSW (3,KEY3)
ISPAR(1)=KEY3
RETURN
END
// DUP
*STORE WS UA FINIS

```

```

FINI0540
FINI0550
FINI0560
FINI0570
FINI0580
FINI0590
FINI0600
FINI0610
FINI0620
FINI0630
FINI0640
FINI0650
FINI0660
FINI0670
FINI0680
FINI0690
FINI0700
FINI0710
FINI0720
FINI0730
FINI0740
FINI0750
FINI0760
FINI0770
FINI0780
FINI0790
FINI0800
FINI0810
FINI0820
FINI0830
FINI0840
FINI0850
FINI0860
FINI0870
FINI0880
FINI0890
FINI0900
FINI0910

```



```

C          IF (NPLOT-1) 2,1,1
C
C          DETERMINE IF GRID WAS DESIRED.
C
C          ***** READ ID. INFO. AND PLOT GRID
C          *****
C          GRID REQUESTED.
C          READ IN THREE LINES OF INFO. FROM KEYBOARD
C          FOR GRAPH LABELS.
C
C          1 READ (6,2000)
C          2000 FORMAT ('
C          READ (6,2001)
C          2001 FORMAT ('
C          READ (6,2002)
C          2002 FORMAT ('
C
C          SET GRAPH SCALE TO UNITY AND INDICATE PEN IS
C          PRESENTLY AT ORIGIN.
C          CALL SCALE (1,1,0,0,0)
C          SET CHARACTER ROTATION VARIABLE FOR
C          HORIZONTAL PRINTING (ALONG X-AXIS).
C          TH=0.
C
C          SERIAL INFO. AND GRAPH LABELS WILL BE PRINTED
C          OUT STARTING 6 INCHES TO THE LEFT AND 2 INCHES
C          BELOW THE ORIGIN. EACH SUCCEEDING LINE WILL
C          BE 0.5 INCHES BELOW THE PREVIOUS ONE. THE
C          CHARACTERS WILL BE THE MINIMUM SIZE ALLOWED
C          (.10 INCHES SQUARE).
C          CALL FCHAR (-6.0,-2.0,10,10,TH)
C          WRITE (7,2003) SER
C          2003 FORMAT (10X,'SERIAL NUMBER--',F10.0)
C          CALL FCHAR (-6.0,-2.5,10,10,TH)
C          WRITE (7,2000)
C          CALL FCHAR (-6.0,-3.0,10,10,TH)
C          WRITE (7,2001)
C          CALL FCHAR (-6.0,-3.5,10,10,TH)
C          WRITE (7,2002)
C          RETURN PEN TO ORIGIN IN THE UP-POSITION.
C          CALL FPLOT (0,0,0,0)
C          PLOT OUT GRID AND LABEL AXES.
C          CALL GRID
C
C          ***** PLOT CURVE FOR EACH VARIABLE INDICATED IN TURN *****
C
C          LOOP THROUGH NUMBER OF VARIABLES TO BE PLOTTED.
C          2 DO 100 JJ=1,NOVAR
C          SET SWITCH SO PEN WILL BE MOVED TO THE FIRST
C          VALUE TO BE PLOTTED IN THE UP-POSITION AND
C          THEN LOWERED TO BEGIN PLOT.
C          L=1
C          SET DISK FILE RECORD INDEX TO FIRST RECORD.
C          :FLO=1
C          STORE LOOP INDEX FOR USE BY SUBROUTINE 'VAPLO'.
C          NIP=JJ
C
C          GRAP1080
C          GRAP1090
C          GRAP1100
C          GRAP1110
C          GRAP1120
C          GRAP1130
C          GRAP1140
C          GRAP1150
C          GRAP1160
C          GRAP1170
C          GRAP1180
C          GRAP1190
C          GRAP1200
C          GRAP1210
C          GRAP1220
C          GRAP1230
C          GRAP1240
C          GRAP1250
C          GRAP1260
C          GRAP1270
C          GRAP1280
C          GRAP1290
C          GRAP1300
C          GRAP1310
C          GRAP1320
C          GRAP1330
C          GRAP1340
C          GRAP1350
C          GRAP1360
C          GRAP1370
C          GRAP1380
C          GRAP1390
C          GRAP1400
C          GRAP1410
C          GRAP1420
C          GRAP1430
C          GRAP1440
C          GRAP1450
C          GRAP1460
C          GRAP1470
C          GRAP1480
C          GRAP1490
C          GRAP1500
C          GRAP1510
C          GRAP1520
C          GRAP1530
C          GRAP1540
C          GRAP1550
C          GRAP1560
C          GRAP1570
C          GRAP1580
C          GRAP1590
C          GRAP1600
C          GRAP1610

```


C	CALL VAPLO	RETRIEVE VARIABLE SCALE FACTOR INTO 'SY'.	GRAP1620
C		SET SCALE FACTORS FOR VARIABLE ('SX' WAS	GRAP1630
C		READ IN).	GRAP1640
C	CALL SCALF (SX,SY,0,0,0)		GRAP1650
C		RETRIEVE NEXT VALUE OF VARIABLE INTO 'VAR' AND	GRAP1660
C		CORRESPONDING TIME VALUE INTO 'CT'.	GRAP1670
C	5 CALL DATA		GRAP1680
C		CHECK FOR FIRST VALUE OF VARIABLE.	GRAP1690
C	GO TO (10,20),L		GRAP1700
C		FIRST VALUE OF VARIABLE INDICATED. MOVE PEN TO	GRAP1710
C		LOCATION IN UP-POSITION LOWERING PEN AFTER	TOGRAP1720
C		MOVEMENT.	GRAP1730
C	10 CALL FPLOT (-2,CT,VAR)		GRAP1740
C		SET SWITCH SO FUTURE VALUES WILL BE PLOTTED	GRAP1750
C		CONTINUOUSLY.	GRAP1760
C	L:2		GRAP1770
C	GO TO 5	RETRIEVE NEXT VALUE OF VARIABLE.	GRAP1780
C		PLOT NEW VALUE CONTINUING GRAPH FROM PREVIOUS	GRAP1790
C		POINT.	GRAP1800
C	20 CALL FPLOT (0,CT,VAR)		GRAP1810
C		CHECK IF GRAPH HAS BEEN COMPLETED.	GRAP1820
C	IF (CT-TEND) 50,50,60		GRAP1830
C		CHECK IF CHANGE IN SCALE IS NEEDED.	GRAP1840
C	50 IF (CT-19.90) 5,55,55		GRAP1850
C		REDUCE X-AXIS SCALE VALUE BY A FACTOR OF TEN.	GRAP1860
C	SXX=.1*SX		GRAP1870
C	XD=CT/.1	INCREASE LAST TIME VALUE BY A FACTOR OF TEN.	GRAP1880
C		COMPUTE NEW STARTING POINT FOR REMAINDER	GRAP1890
C		OF GRAPH.	GRAP1900
C	XDARN=XD-CT		GRAP1910
C		SET NEW SCALE FACTORS AND INDICATE PRESENT	GRAP1920
C		LOCATION OF PEN IS ORIGIN.	GRAP1930
C	CALL SCALF (SXX,SY,XD,VAR)		GRAP1940
C		RETRIEVE NEXT VALUE OF VARIABLE INTO 'VAR' AND	GRAP1950
C		CORRESPONDING TIME VALUE INTO 'CT'.	GRAP1960
C	CALL DATA		GRAP1970
C		COMPUTE MODIFIED TIME VALUE BECAUSE OF NEW	GRAP1980
C		X-AXIS SCALE FACTOR AND ORIGIN.	GRAP1990
C	CCT=XDARN+CT		GRAP2000
C		PLOT NEW VALUE CONTINUING GRAPH FROM PREVIOUS	GRAP2010
C		POINT.	GRAP2020
C	CALL FPLOT (0,CCT,VAR)		GRAP2030
C		CHECK IF GRAPH HAS BEEN COMPLETED.	GRAP2040
C	IF (CT-TEND) 57,57,57		GRAP2050
C		SET CHARACTER ROTATION VALUE FOR	GRAP2060
C		HORIZONTAL PRINTING (ALONG X-AXIS).	GRAP2070
C	58 TH=0.0		GRAP2080
C		POSITION PEN AND SET CHARACTER SIZE FOR	GRAP2090
C		PRINTING OF LABEL.	GRAP2100
C			GRAP2110
C			GRAP2120
C			GRAP2130
C			GRAP2140
C			GRAP2150


```

VD=.05/SY
CALL FCHAR (CCT+XD,VAR-YD,.10,.10,TH)
GO TO 59
C
C
C      SET CHARACTER ROTATION VALUE FOR
      HORIZONTAL PRINTING (ALONG X-AXIS).
C
C      60 TH=0.0
C
C      XD=.05/SX
      YD=.05/SY
      CALL FCHAR (CT+XD,VAR-YD,.10,.10,TH)
C
C      PRINT OUT APPROPRIATE LABEL.
C
C      59 CALL PRINT
      RAISE PEN AND RETURN IT TO ORIGIN.
C      CALL FPLOT (1.0,.0.)
      LOOP IF MORE VARIABLES ARE TO BE PLOTTED.
C      100 CONTINUE
C
C***** RELOAD AND EXECUTE 'GENIS' TO START NEW PROBLEM *****
C
C      ALL PLOTTING DONE. LOAD AND EXECUTE 'GENIS'
      MAINLINE TO START NEW PROBLEM.
C
C      CALL LINK (GENIS)
C
C      END
// DUP
*STORE      WS UA GRAPH

```

```

GRAP22160
GRAP22170
GRAP22180
GRAP22190
GRAP22200
GRAP22210
GRAP22220
GRAP22230
GRAP22240
GRAP22250
GRAP22260
GRAP22270
GRAP22280
GRAP22290
GRAP22300
GRAP22310
GRAP22320
GRAP22330
GRAP22340
GRAP22350
GRAP22360
GRAP22370
GRAP22380
GRAP22390
GRAP22400
GRAP22410
GRAP22420
GRAP22430
GRAP22440
GRAP22450

```



```

C          PRINT OUT LABEL.
C          WRITE (7,5) X
C          FORMAT (F5.1)
C          5
C          UPDATE LABEL
C          20
C          X=X+1.
C          C***** DETERMINE IF TIME AXIS EXTENDS BEYOND 20 SECONDS AND FIND
C          C***** END POINT IF SO.
C          CALL FPLLOT (-2.0,0.0.)
C          MOVE PEN TO ORIGIN AND LOWER IT.
C          CHECK IF NECESSARY TO EXTEND X-AXIS BEYOND
C          20 SECONDS.
C          TEST=TEND-20.
C          IF (TEST) 100,100,200
C          X-AXIS MUST BE EXTENDED. COMPUTE FINAL VALUE
C          TRUNCATE TO INTEGER
C          200 TEST1=TEST/10.0+1.0
C          ITTEST=TEST1
C          C***** PLOT GRID FOR REMAINDER OF TIME AXIS.
C          SXX=.1*SX
C          REDUCE X-AXIS SCALE VALUE BY A FACTOR OF 10.
C          SET NEW SCALE FACTORS.
C          CALL SCALF (SXX,1.0,0.0.)
C          PLOT REMAINDER OF X-AXIS GRID.
C          CALL FGRID (0,200.0,0.10,ITEST)
C          COMPUTE VALUE TO CENTER X-AXIS LABELS ON
C          TICK MARKS.
C          XD=.05/SXX
C          SET INITIAL LABEL.
C          X=30.
C          LOOP TO PRINT REMAINING LABELS.
C          DO 11 I=1,ITEST
C          MOVE PEN TO APPROPRIATE POSITION.
C          CALL FCHAR (180,X-XD,MYE,10,10,TH)
C          PRINT LABEL.
C          WRITE (7,5) X
C          UPDATE LABEL.
C          11 X=X+10.
C          C***** PLOT GRID FOR +Y AND -Y AXES.
C          MOVE PEN TO ORIGIN AND LOWER IT.
C          CALL FPLLOT (-2.0,0.0.)
C          RESET ORIGINAL SCALE FACTORS.
C          100 CALL SCALF (SX,1.0,0.0.)
C          PLOT GRID FOR +Y DIRECTION
C          CALL FGRID(1.0,0.0,-5,10)
C          PLOT GRID FOR -Y DIRECTION

```

```

GRID0540
GRID0550
GRID0560
GRID0570
GRID0580
GRID0590
GRID0600
GRID0610
GRID0620
GRID0630
GRID0640
GRID0650
GRID0660
GRID0670
GRID0680
GRID0690
GRID0700
GRID0710
GRID0720
GRID0730
GRID0740
GRID0750
GRID0760
GRID0770
GRID0780
GRID0790
GRID0800
GRID0810
GRID0820
GRID0830
GRID0840
GRID0850
GRID0860
GRID0870
GRID0880
GRID0890
GRID0900
GRID0910
GRID0920
GRID0930
GRID0940
GRID0950
GRID0960
GRID0970
GRID0980
GRID0990
GRID1000
GRID1010
GRID1020
GRID1030
GRID1040
GRID1050
GRID1060
GRID1070

```



```

C      CALL FGRID (3.0,0.0,5.10)
C      SET CHARACTER ROTATION VALUE FOR
C      HORIZONTAL (ALONG X-AXIS)
C
C      TH=0.0
C
C      SET VALUE TO START PRINTING Y-AXIS LABELS 0.77
C      INCHES TO THE LEFT OF THE ORIGIN.
C
C      X=-0.77/SX
C
C      INITIALIZE Y-AXIS LABEL
C      SET VALUE FOR CENTERING OF LABELS ON TICK
C      MARKS.
C
C      WYE=5.
C      XD=.05
C
C      DO 20 I=1,11
C      .CALL FCHAR (X,WYE-XD,.10,.10,TH)
C      WRITE(7,5) WYE
C      WYE=WYE-1.
C      20
C
C      ***** PLOT TITLE ON TIME AXIS.
C
C      SET CHARACTER ROTATION VALUE FOR
C      HORIZONTAL.
C
C      TH=0.
C
C      MOVE PEN TO LOCATION FOR PRINTING X-AXIS
C      TITLE.
C      CALL FCHAR (3.0,-1.25,.2,.2,TH)
C      PRINT TITLE.
C      WRITE (7,6)
C      FORMAT ('TIME(SECS):')
C
C      ***** PLOT SCALE FACTORS FOR ALL THE VARIABLES BEING PLOTTED.
C
C      SET CHARACTER ROTATION VALUE FOR VERTICAL
C      BOTTOM TO TOP PRINTING.
C
C      TH=PI/2.
C
C      X=-1.0/SX
C
C      DO 69 NIP=1,NOVAR
C      CALL VAPLO
C      WKE=NVAR(NIP)
C
C      MOVE PEN TO BEGINNING OF LINE. CHARACTERS
C      WILL BE OF SIZE 0.2 INCHES SQUARE.
C      CALL FCHAR (X,-1.5,.2,.2,TH)
C      COMPUTE PLOT SCALE FOR VARIABLE.
C      SY=1./SY
C
C      GO TO (30,40,50,60,70,80,90,92,94,96,98,101), WKE
C
C      BRANCH ACCORDING TO VARIABLE IN QUESTION.

```

GRID1080
 GRID1090
 GRID1100
 GRID1110
 GRID1120
 GRID1130
 GRID1140
 GRID1150
 GRID1160
 GRID1170
 GRID1180
 GRID1190
 GRID1200
 GRID1210
 GRID1220
 GRID1230
 GRID1240
 GRID1250
 GRID1260
 GRID1270
 GRID1280
 GRID1290
 GRID1300
 GRID1310
 GRID1320
 GRID1330
 GRID1340
 GRID1350
 GRID1360
 GRID1370
 GRID1380
 GRID1390
 GRID1400
 GRID1410
 GRID1420
 GRID1430
 GRID1440
 GRID1450
 GRID1460
 GRID1470
 GRID1480
 GRID1490
 GRID1500
 GRID1510
 GRID1520
 GRID1530
 GRID1540
 GRID1550
 GRID1560
 GRID1570
 GRID1580
 GRID1590
 GRID1600
 GRID1610


```

C      30 WRITE (7,31) SYI
      31 FORMAT('AI(1 IN.,F7.1,' AMPS)')
      GO TO 69
C      40 WRITE (7,41) SYI
      41 FORMAT('EB(1 IN.,F5.1,' VOLTS)')
      GO TO 69
C      50 WRITE (7,51) SYI
      51 FORMAT('NPM(1 IN.,F5.1,' RPM)')
      GO TO 69
C      60 WRITE (7,61) SYI
      61 FORMAT('EVI(1 IN.,F5.1,' AMPS)')
      GO TO 69
C      70 WRITE (7,71) SYI
      71 FORMAT('NP(1 IN.,F5.1,' RPM)')
      GO TO 69
C      80 WRITE (7,81) SYI
      81 FORMAT('KW(1 IN.,F6.1,' KW)')
      GO TO 69
C      90 WRITE (7,91) SYI
      91 FORMAT('VK(1 IN.,F4.1,' KTS)')
      GO TO 69
C      92 WRITE (7,93) SYI
      93 FORMAT('OP(1 IN.,F8.1,' FT-LBS)')
      GO TO 69
C      94 WRITE (7,95) SYI
      95 FORMAT('TP(1 IN.,F9.1,' LBS)')
      GO TO 69
C      96 WRITE (7,97) SYI
      97 FORMAT('XS(1 IN.,F6.1,' FEET)')
      GO TO 69
C      98 WRITE (7,99) SYI
      99 FORMAT('OI(1 IN.,F8.1,' FT-LBS)')
      GO TO 69
C     101 WRITE (7,102) SYI
     102 FORMAT('TEMP(1 IN.,F5.1,' DEG-CENT)')
C
C      69 X=X-(.4/SX)
C
C***** MOVE PEN TO ORIGIN IN THE UP-POSITION AND RETURN.
      COMPUTE POSITION FOR NEXT LINE TO BE 0.4 INCHES
      FURTHER LEFT OF ORIGIN THAN PREVIOUS LINE.

```

```

GRID1620
GRID1630
GRID1640
GRID1650
GRID1660
GRID1670
GRID1680
GRID1690
GRID1700
GRID1710
GRID1720
GRID1730
GRID1740
GRID1750
GRID1760
GRID1770
GRID1780
GRID1790
GRID1800
GRID1810
GRID1820
GRID1830
GRID1840
GRID1850
GRID1860
GRID1870
GRID1880
GRID1890
GRID1900
GRID1910
GRID1920
GRID1930
GRID1940
GRID1950
GRID1960
GRID1970
GRID1980
GRID1990
GRID2000
GRID2010
GRID2020
GRID2030
GRID2040
GRID2050
GRID2060
GRID2070
GRID2080
GRID2090
GRID2100
GRID2110
GRID2120
GRID2130
GRID2140
GRID2150

```


GR102160
GR102170
GR102180
GR102190
GR102200
GR102210
GR102220

C CALL FPLOT (0,0,0,0.)
RETURN
C
// DUP
*STORE WS UA GRID


```

// FOR
*LIST ALL
*ONE WORD INTEGERS
SUBROUTINE DATA
VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968

COMMON RO,PI,IZB,DAR,PDR,D,XJPA,XV,XJSH,XJD,XJG,XJG,
1 XJVP,XJGG,FIWAX,RA,RG,CI,XK1,QR,FR,CWM,PCMS,XLA,
2 TAU1,TAU2,XK6,ALP1,ALP2,ALP3,ALP4,SS
COMMON NOVAR,NVAR(10),JJ,ICNTR,NPL,DATA,NOP,NOM,NIP
COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12
COMMON DT,CT,T,TEND,TINSC
COMMON Z,Y,X,NPV,FGI,FIP,XNP,VS,XS,SIE,AI,RV,TEVP
COMMON CZ,CY,CX,NPM,CFG1,CFIP,CXNP,CVS,CXS,CSIE,CAICRV,CTEMP
COMMON ZDC,YDC,YDD,XNPM,FGIDO,FIPDO,XNPDO,VSDO,XSDO,SIEDO,AIDO,RVDO,
1 SWDO
COMMON XNE,XNEDO,VA,THETO,THET1,THETA,THETD,THEIC,ODE,OG,EG,ES,EC,
10,OF,AV,AIHAT,OP,QPI,QI,TP,XPROI,RAVAR,XKW,CVK
COMMON CFG,SPGEN,CFGA(13),TRNFG,ZGP,RFG,XLFG,TAUFG,TRNSF,RX,XM,XLA,
15,AVP,R,XGEN
COMMON CFM,SPNOT,CFMA(13),TRNFM,ZVP,RFM,XLFM,TAUFM,FMRV,TAUA,TAUB,
COMMON B(64),RB(64),E(32),EE(32)
COMMON EHP(20),WDED,WDED(20),THDED,THDED(20),XKK
COMMON IFLO,JELO,KELO,LELO,WELO
COMMON SPARE(5),ISPAR(5)

READ(1,'IELO')CT,AI,EB,RPMPV,FMI,RPMP,XKK,CVK,QPI,XPROI,CXS
1,QI,TEVP

READ LATEST VALUES OF VARIABLES FROM DISK
USING CURRENT VALUE OF 'IELO' AS INDEX. 'IELO'
WILL BE UPDATED TO NEXT RECORD AUTOMATICALLY.

STORE NUMBER OF VARIABLE WE ARE DEALING WITH
AND BRANCH ACCORDINGLY TO RETRIEVE ITS VALUE.

NOOK=NVAR(JJ)
GO TO (20,20,30,40,50,60,70,80,90,100,110,120), NOOK

VARIABLE 'AI' - ARM. CIR. CUR.
VARIABLE 'EB' - BUS VOLTS
VARIABLE 'RPMPH' - DIESEL SPEED
VARIABLE 'FMI' - MOTOR FIELD CUR.

VAR=AI
RETURN
20
VAR=EB
RETURN
30
VAR=RPMPV
RETURN
40
VAR=FMI
RETURN

```


SDAT0540
SDAT0550
SDAT0560
SDAT0570
SDAT0580
SDAT0590
SDAT0600
SDAT0610
SDAT0620
SDAT0630
SDAT0640
SDAT0650
SDAT0660
SDAT0670
SDAT0680
SDAT0690
SDAT0700
SDAT0710
SDAT0720
SDAT0730
SDAT0740
SDAT0750
SDAT0760
SDAT0770
SDAT0780
SDAT0790
SDAT0800
SDAT0810
SDAT0820
SDAT0830
SDAT0840
SDAT0850
SDAT0860
SDAT0870
SDAT0880
SDAT0890

VARIABLE 'RPM' - PROP. SPEED

VARIABLE 'KW' - POWER

VARIABLE 'CVK' - VESSEL SPEED

VARIABLE 'CPI' - PROP. TORQUE

VARIABLE 'XPROI' - PROP. THRUST

VARIABLE 'CXS' - DISTANCE

VARIABLE 'CI' - ICE TORQUE

VARIABLE 'TEMP' - MOTOR TEMP.

C 50 VAR=RPM
RETURN
C 60 VAR=KW
RETURN
C 70 VAR=CVK
RETURN
C 80 VAR=CPI
RETURN
C 90 VAR=XPROI
RETURN
C 100 VAR=CXS
RETURN
C 110 VAR=CI
RETURN
C 120 VAR=TEMP
RETURN
C
// DUP
*STORE WS UA DATA
END


```

// FOR
*LIST ALL
*ONE WORD INTEGERS
SUBROUTINE PRINT
VERSION 1 MODIFICATION 0 - DATED MARCH 1, 1968

COMMON RO,PI,IZR,DAR,POR,D,XUPA,XJM,XJSH,XJD,XJG,EEG,
1 XJPD,XJGG,FVAX,RA,RG,CI,XK1,CBRG,CSEH,CWM,PCMS,XLA,
2 TAU1,TAU2,XK5,ALP1,ALP2,ALP3,ALP4,XVASS
COMMON NOVAP,IVAR(10),JJ,ICNTR,NOLOT,NDATA,NOP,MON,NIP
COMMON SX,SY,SY1,SY2,SY3,SY4,SY5,SY6,SY7,SY8,SY9,SY10,SY11,SY12
COMMON DT,CT,T,TEND,TMSC
COMMON ZY,XNPV,FGI,FIP,XNP,VS,XS,CSIE,AI,RY,TEMP
COMMON CZ,CY,CXNPM,CFG1,CFG2,CXNP,CVS,CXS,CSIE,CAI,CRV,CTEMP
COMMON ZDO,YDO,XNPVD,FGID0,FID0,XIPDO,VSDO,XSDO,SIEDO,ALDO,RVDO,TPRINO150
1 EVD0
COMMON XNS,XNEDO,VA,THETO,THET1,THET2,THETD,THEIC,QDE,CG,EG,ER,EC,PRINO170
10X,CEPM,AIMAT,CP,OPI,CI,TP,XPRO,PR,VAR,XKM,CVK
COMMON CFG,SPGEN,CFGAL(13),TRIEG,ZOP,REG,XLEU,TAUFG,TRNSF,RX,XV,XLA,PRINO180
135,AVSTR,XGEN
COMMON CEV,SPNOT,CFNA(13),TRNEN,ZUP,REM,XLFM,TAUFV,FNIRV,TAUA,TAUB,PRINO210
COMMON S(54),R5(64),E5(32)
COMMON ENO(20),WDED,WDED(20),TDED,THDED(20),XKX
COMMON IEL0,JELO,KELO,LELO,VELO
COMMON SPARE(5),ISPAR(5)

VIKE=VAR(UJ)
GO TO (10,20,30,40,50,60,70,80,90,100,110,120), MIKE

10 WRITE (7,11)
11 FORMAT ('----ARM. CIR. CUR. ')
RETURN
20 WRITE (7,21)
21 FORMAT ('----BUS VOLTS')
RETURN
30 WRITE (7,31)
31 FORMAT ('----DIESEL SPEED')
RETURN
40 WRITE (7,41)
41 FORMAT ('----MOTOR FIELD CUR. ')
RETURN

```



```

C      50 WRITE (7,51)      VARIABLE 'RPM' - PROP. SPEED
      51 FORMAT ('----PROP SPEED')
      RETURN

C      60 WRITE (7,61)      VARIABLE 'KW' - POWER
      61 FORMAT ('----POWER')
      RETURN

C      70 WRITE (7,71)      VARIABLE 'CVK' - VESSEL SPEED
      71 FORMAT ('----VESSEL SPEED')
      RETURN

C      80 WRITE (7,81)      VARIABLE 'QPI' - PROP. TORQUE
      81 FORMAT ('----PROP TORQUE')
      RETURN

C      90 WRITE (7,91)      VARIABLE 'XPROI' - PROP. THRUST
      91 FORMAT ('----PROP THRUST')
      RETURN

C     100 WRITE (7,101)     VARIABLE 'CXS' - DISTANCE
     101 FORMAT ('----DISTANCE')
     RETURN

C     110 WRITE (7,111)     VARIABLE 'QI' - ICE TORQUE
     111 FORMAT ('----ICE TORQUE')
     RETURN

C     120 WRITE (7,121)     VARIABLE 'TEMP' - MOTOR TEMP.
     121 FORMAT ('----MOTOR TEMP.')
     RETURN

C      // DUP
      *STORE WS DA PRINT
      END

```

```

PRINC540
PRINC550
PRINC560
PRINC570
PRINC580
PRINC590
PRINC600
PRINC610
PRINC620
PRINC630
PRINC640
PRINC650
PRINC660
PRINC670
PRINC680
PRINC690
PRINC700
PRINC710
PRINC720
PRINC730
PRINC740
PRINC750
PRINC760
PRINC770
PRINC780
PRINC790
PRINC800
PRINC810
PRINC820
PRINC830
PRINC840
PRINC850
PRINC860
PRINC870
PRINC880
PRINC890
PRINC900
PRINC910
PRINC920
PRINC930
PRINC940
PRINC950
PRINC960

```



```

C 60 SY=SY6
RETURN
C
C 70 SY=SY7
RETURN
C
C 80 SY=SY8
RETURN
C
C 90 SY=SY9
RETURN
C
C 100 SY=SY10
RETURN
C
C 110 SY=SY11
RETURN
C
C 120 SY=SY12
RETURN
C
// SUP
*STORE WS UA VAPLO
END

```

VARIABLE 'XKW'	- POWER	VAPLO540
VARIABLE 'CVK'	- VESSEL SPEED	VAPLO550
VARIABLE 'GPI'	- PROP. TORQUE	VAPLO560
VARIABLE 'XPROT'	- PROP. THRUST	VAPLO570
VARIABLE 'CXS'	- DISTANCE	VAPLO580
VARIABLE 'GI'	- ICE TORQUE	VAPLO590
VARIABLE 'ITEMD'	- MOTOR TEMP.	VAPLO600
		VAPLO610
		VAPLO620
		VAPLO630
		VAPLO640
		VAPLO650
		VAPLO660
		VAPLO670
		VAPLO680
		VAPLO690
		VAPLO700
		VAPLO710
		VAPLO720
		VAPLO730
		VAPLO740
		VAPLO750
		VAPLO760
		VAPLO770
		VAPLO780
		VAPLO790
		VAPLO800
		VAPLO810
		VAPLO820
		VAPLO830
		VAPLO840

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) U. S. Coast Guard Headquarters 1300 E Street, N. W. Washington, D. C. 20591		2a. REPORT SECURITY CLASSIFICATION Unclassified	
		2b. GROUP None	
3. REPORT TITLE Ship Design and Maintenance Computer Program Icebreaker Propulsion System Simulation (Diesel-Electric)			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates)			
5. AUTHOR(S) (First name, middle initial, last name) Jack W. Lewis			
6. REPORT DATE May 1968		7a. TOTAL NO. OF PAGES 257	7b. NO. OF REFS 3
8a. CONTRACT OR GRANT NO.		9a. ORIGINATOR'S REPORT NUMBER(S) ENE-11	
b. PROJECT NO. 1702 Replacement Icebreakers			
c.		9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
d.			
10. DISTRIBUTION STATEMENT This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of Commandant(E), U.S. Coast Guard Headquarters, 1300 E St., N.W., Washington, D. C. 20591			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY U. S. Coast Guard	

13. ABSTRACT

This document describes a digital computer program written for the IBM 1130 Computing System that is capable of simulating transient conditions within a diesel-electric propulsion system typical of installations aboard most icebreaker type vessels. The program will simulate various conditions to which an icebreaker may be subjected. These include "crash reversals", acceleration under free route or ice clogged channel conditions, ice ramming and subjection of the propellers to ice encounters. The program accomplishes the simulation by solving numerically the set of simultaneous non-linear differential equations which describe the dynamic behavior of the complete propulsion system including propeller and ship motion dynamics. At the option of the user, output data can be plotted.

14.	KEY WORDS	LINK A		LINK B		LINK C	
		ROLE	WT	ROLE	WT	ROLE	WT
	Icebreakers						
	Ship Design						
	IBM 1130						
	Computer Program						
	Simulation						
	Propulsion Control						
	Automated Control Systems						
	Propulsion Plants						
	Biesels						
	Electric Drive						
	Propellers						